Rainbow Trees in Edge-Colored Complete Graphs and Block Decompositions of Almost Complete Graphs

by

Katherine E. Perry

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 7, 2017

Keywords: edge-coloring, complete graph, rainbow spanning tree, balanced sampling plan, two dimensions, contiguous

Approved by

Chris Rodger, Chair, Don Logan Endowed Chair in Mathematics, Professor of Mathematics
Dean Hoffman, Professor of Mathematics
Peter Johnson, Professor of Mathematics
Jessica McDonald, Assistant Professor of Mathematics
Michel Smith, Professor of Mathematics

Abstract

This dissertation focuses on two problems, the first involving the existence of many edge-disjoint rainbow spanning trees in edge-colored complete graphs, and the second, creating a balanced sampling plan for a two-dimensional array, excluding contiguous units.

A spanning tree of a properly edge-colored complete graph, K_n, is rainbow provided that no two different edges in the tree bear the same color. In 1996, Brualdi and Hollingsworth conjectured that if K_{2m} is properly $(2m-1)$-edge-colored, then the edges of K_{2m} can be partitioned into m rainbow spanning trees except when $m = 2$. The existence of $\lfloor m/(500 \log(2m)) \rfloor$ mutually edge-disjoint spanning trees in the case that $m \geq 500,000$ was recently proved using probabilistic techniques. By means of an explicit, constructive approach, we construct $\lfloor \sqrt{6m+9}/3 \rfloor$ mutually edge-disjoint rainbow spanning trees for any positive value of m. Not only are the rainbow trees produced, but also some structure of each rainbow spanning tree is determined in the process. This improves upon best constructive result to date in the literature which produces exactly three rainbow trees. It also improves upon the probabilistic result for all m at most 5.7×10^7.

Balanced sampling plans excluding contiguous units (BSECs) were first introduced by Hedayat, Rao, and Stufken in 1988. The idea of generalizing this definition to two dimensions was first formalized by Bryant, Chang, Rodger, and Wei in 2002 where the case for block size 3 and $\lambda = 1$ (the number of blocks each pair of points appears in together) was completely solved. These designs are useful for items arranged in a two-dimensional array where contiguous units provide similar information. In this dissertation, a complete solution for the existence of two-dimensional BSECs with block size 3 and $\lambda = 3$ is provided.

Acknowledgments

First, I would like to thank my adviser, Dr. Chris Rodger, for his never-failing support, direction, kindness, and patience. He always allowed me the space and time to figure things out for myself, turned me into the teacher that I am, and spent months searching for the perfect research problem ("Any design theory will be by accident!"). I knew within one week of his graph theory class my first semester that I wanted him to be my adviser and will forever be grateful he agreed.

I would also like to thank the members of my committee for their encouragement and suggestions throughout this process and for the following: Dr. Dean Hoffman for welcoming me into any room and opening his home each year for the Combinatorics party; Dr. Jessica McDonald for her wonderful advice and always encouraging me to apply to something every time I found myself in her office; Dr. Michel Smith for being on my committee simply because I asked; Dr. James Cross for his corrections and incredibly impressive memory of students from years past; and Dr. Pete Johnson for always attending my talks even when he's seen them before, taking me to the SAMSA-Masamu program in Africa, and welcoming me at Friday seminars.

Next, I would like to thank my friends who make up my Auburn Family. I am fortunate enough that there are too many people to list names, but y'all know who you are. The Auburn Family is real and I am honored to be a part of it.

I would also like to thank my Perry family: Georgia, Bob, Rick, Teddy, Sarah, Nydia, Miranda, and Penelope.

And finally, if the reader will permit me, I wish to dedicate this dissertation to my mother, Gail, who would love me with or without it.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This dissertation focuses on two problems, the first involving the existence of many edge-disjoint rainbow spanning trees in edge-colored complete graphs, and the second, creating a balanced sampling plan for a two-dimensional array, excluding contiguous units. Each problem will be introduced in turn in Chapter 1. Chapter 2 will focus on the first problem and the second problem will be the subject of Chapter 3. In Chapter 4 a discussion of open problems related to each is provided.

1.1 Rainbow Spanning Trees in Edge-Colored Complete Graphs

A spanning tree T of a graph G is an acyclic connected subgraph of G for which $V(T) = V(G)$. A proper k-edge-coloring of a graph G is a mapping from $E(G)$ into a set of colors, $\{1, 2, ..., k\}$, such that adjacent edges of G receive distinct colors. Since all edge-colorings considered in this dissertation are proper, if G has a proper k-edge-coloring, then G is said to be k-edge-colored. The chromatic index $\chi'(G)$ of a graph G is the minimum number k such that G is k-edge-colorable. It is well known that $\chi'(K_{2m}) = 2m - 1$ and thus, if K_{2m} is properly $(2m - 1)$-edge-colored, each color appears on exactly one edge at each vertex.

A subgraph in an edge-colored graph is said to be rainbow (sometimes called multicolored or poly-chromatic) if its edges receive distinct colors. It is not hard to see that with any $(2m - 1)$-edge-coloring of K_{2m}, a rainbow spanning tree can be found by taking the spanning star, S_v, with any center $v \in V(K_{2m})$. Further, K_{2m} has $m(2m - 1)$ edges and it is well known that these edges can be partitioned into m spanning trees. This led Brualdi and Hollingsworth [4] to make the following conjecture in 1996.

Conjecture 1.1 ([4]). *If K_{2m} is $(2m-1)$-edge-colored, then the edges of K_{2m} can be partitioned into m rainbow spanning trees except when $m = 2$.*

Based on Brualdi and Hollingsworth's concept, Constantine [8] proposed two related conjectures in 2002.

Conjecture 1.2 ([8], Weak version). *K_{2m} can be edge-colored with $2m-1$ colors in such a way that the edges can be partitioned into m isomorphic rainbow spanning trees except when $m = 2$.*

Conjecture 1.2 was proved to be true by Akbari, Alipour, Fu, and Lo in 2006 [1].

Conjecture 1.3 ([8], Strong version). *If K_{2m} is $(2m-1)$-edge-colored, then the edges of K_{2m} can be partitioned into m isomorphic rainbow spanning trees except when $m = 2$.*

Concerning Conjecture 1.1, in [4], Brualdi and Hollingsworth proved that there exist two edge-disjoint rainbow spanning trees for $m > 2$, and in 2000, Krussel, Marshall, and Verrall [15] improved this result to three spanning trees. Recently, Carraher, Hartke, and Horn [6] showed that if m is sufficiently large ($m \geq 500,000$) then an edge-colored K_{2m} in which each color appears on at most m edges contains at least $\left\lfloor \frac{m}{500\log(2m)} \right\rfloor$ edge-disjoint rainbow spanning trees.

Essentially, not much has been done on Conjecture 1.3. The best result so far is by Fu and Lo [10]. They proved that three isomorphic rainbow spanning trees exist in any $(2m-1)$-edge-colored K_{2m} for each $m \geq 14$.

In this dissertation, we focus on Conjecture 1.1 by proving that in any $(2m-1)$-edge-coloring of K_{2m}, $m \geq 1$, there exist at least $\left\lfloor \frac{\sqrt{6m+9}}{3} \right\rfloor$ mutually edge-disjoint rainbow spanning trees. Asymptotically, this is not as good as the bound in [6], but our result applies to all values of m and it is better until m is extremely large (over 5.7×10^7). Furthermore, instead of using the non-constructive probabilistic method to prove the result, as was used in [6], we derive our bound by means of an explicit, constructive approach. So, not only do we actually produce the rainbow trees, but also some structure of each rainbow spanning tree is determined in the process. It should be noted that the current best constructive result (before ours) is the one in the paper by Krussel, Marshall, and Verrall [15] which produces just three rainbow spanning trees. Here is our main result.

Theorem 1.1. *Let K_{2m} be a properly $(2m-1)$-edge-colored graph. Then there exist $\Omega_m = \left\lfloor \frac{\sqrt{6m+9}}{3} \right\rfloor$ mutually edge-disjoint rainbow spanning trees, say $T_1, T_2, \ldots, T_{\Omega_m}$, with the following properties.*

(i) Each tree has a designated distinct root.

(ii) The root of T_1 has degree $(2m-1)-2(\Omega_m-1)$ in and has at least $(2m-1)-4(\Omega_m-1)$ adjacent leaves.

(iii) For $2 \leq i \leq \Omega_m$, The root of T_i has degree $(2m-1)-i-2(\Omega_m-i)$ and has at least $(2m-1)-2i-4(\Omega_m-i)$ adjacent leaves.

The proof of this result is also of interest, involving three inductions being applied simultaneously.

It is worth mentioning here that the above conjectures will play important roles in certain applications if they are true. Notice that a rainbow spanning tree is orthogonal to the 1-factorization of K_{2m} (induced by any $(2m-1)$-edge-coloring). An application of parallelisms of complete designs to population genetics data can be found in [3]. Parallelisms are also useful in partitioning consecutive positive integers into sets of equal size with equal power sums [14]. In addition, the discussions of applying colored matchings and design parallelisms to parallel computing appeared in [11].

1.2 Balanced Sampling Designs Excluding Contiguous Units

Balanced sampling designs excluding contiguous units (BSECs) were first introduced by Hedayat, Rao, and Stufken in 1988 [13]. These designs can be used to more efficiently gather data in situations where the units near each other provide similar information. In this instance, the v units are named with elements of $\mathbb{Z}_v$ and are arranged in a one-dimensional array in which two units $i \leq j$ are said to be contiguous if and only if $\mid i - j \mid = 1$ or $\{i, j\} = \{0, v-1\}$.

A one-dimensional k-sized balanced sampling plan excluding contiguous units of order v and index λ, 1-BSEC(v, k, λ), is a pair (X, B), where X is a set of v points, $\mathbb{Z}_v$, and B is a collection of (not necessarily distinct) k-subsets of X (called blocks), such that any two contiguous points do not appear together in any block, while any two noncontiguous points appear together in exactly λ blocks. Constructions of 1-BSECs have been studied in multiple papers [7, 12, 13, 19]. The following complete solution for the existence of 1-BSECs with block size 3 was found by Colbourn and Ling in 1998 [7].

Theorem 1.2. [7] *A 1-BSEC$(v, 3, \lambda)$ exists if and only if either $v \geq 9$ and $\lambda(v-3) \equiv 0$ (mod 6) or $v \in \{1, 3\}$.*

Although the idea of generalizing 1-BSECs to two dimensions was first suggested by Hedayat, Rao, and Stufken in 1988 [13], a formal definition was not given until the paper of Bryant, Chang, Rodger, and Wei in 2002 [5]. This was done by first generalizing the notion of contiguous.

Given a set of points $\mathbb{Z}_m \times \mathbb{Z}_n$ arranged in two dimensions, the 2-contiguous points to a point (x, y) are $(x-1, y)$, $(x+1, y)$, $(x, y+1)$, and $(x, y-1)$, reducing sums mod m and mod n in the first and second coordinates respectively. We also note here that if m or n were allowed to be less than 3, then each point would not have four 2-contiguous points. Thus, in this dissertation when considering this two-dimensional case, we assume that neither m nor n is smaller than 3.

Bryant, Chang, Rodger, and Wei then used this definition of 2-contiguous to generalize 1-BSECs to two dimensions. They defined a 2-BSEC(m, n, k, λ), $m, n \geq 3$, to be a pair (X, B) where $X = \mathbb{Z}_m \times \mathbb{Z}_n$ and B is a collection of k-subsets of X (called blocks) such that each pair of 2-contiguous points do not appear together in any block, while any other two points appear together in exactly λ blocks. It is easy to see that a 2-BSEC(m, n, k, λ) can be thought of as a decomposition of $\lambda(K_{mn} - E(H))$ into K_k's, where H is a subgraph of K_{mn} consisting of edges between 2-contiguous points and each edge in K_{mn} has multiplicity λ.

When it causes no confusion to the reader, we will refer to 2-contiguous points as simply being contiguous. We can also observe here that if we allowed n to equal 1 then a 2-BSEC$(m, 1, k, \lambda)$ is equivalent to a 1-BSEC(m, k, λ).

Our result considers constructions of 2-BSECs in the case where $\lambda = 3$. Before we state our result, we first observe the following necessary conditions for the existence of a 2-BSEC.

Lemma 1.3. [5] *If a 2-BSEC(m, n, k, λ) exists, then*

1. $\lambda mn(mn - 5) \equiv 0$ *(mod $k(k-1)$), and*

2. $\lambda(mn - 5) \equiv 0$ *(mod $k - 1$).*

Proof. Condition (1) follows due to the fact that the number of noncontiguous pairs of points is $\frac{\lambda mn(mn-5)}{2}$ and this number must be divisible by the number of pairs of points in a block, namely $\frac{k(k-1)}{2}$. Condition (2) follows from the fact that for each fixed point (x, y), there are $\lambda(mn - 5)$ noncontiguous points to (x, y), which must be divisible by the number of points in each block other than (x, y), namely $(k - 1)$. $\square$

The existence problem for a 2-BSEC$(m, n, 3, 1)$ was completely solved by Bryant, Chang, Rodger, and Wei in 2002 [5].

Theorem 1.4. [5] *There exists a 2-BSEC$(m, n, 3, 1)$ if and only if m and n are odd and*

1. m *or* $n \equiv 3$ *(mod 6), or*

2. $m \not\equiv n$ *(mod 6).*

Our result extends Theorem 1.4, solving the case where $\lambda = 3$. Here is our main result.

Theorem 1.5. *A 2-BSEC$(m, n, 3, 3)$ exists if and only if m and n are odd.*

It is worth mentioning here that our result was recommended by the referees for publication in the The Australasian Journal of Combinatorics, but shortly after our result was submitted, Wang, Feng, Zhang, and Xu submitted a result encompassing ours [9]. Their paper acknowledges our result in their concluding remarks.

Like 1-BSECs, 2-BSECs also have practical applications. These designs can be used to test small land plots at a dump site for chemical waste, where clearly contiguous plots will give similar information. They can also be used for finite population sampling. Since many species are social creatures, they tend to live in clusters instead of being spread throughout a region. Thus, sampling by excluding contiguous units is much more likely to provide a more accurate population estimate.

Chapter 2

Problem 1: The Number of Edge-Disjoint Rainbow Spanning Trees in Edge-Colored Complete Graphs

In this chapter we will prove Theorem 1.1. For convenience, we restate it here.

Theorem 1.1. *Let K_{2m} be a properly $(2m-1)$-edge-colored graph. Then there exist $\Omega_m = \left\lfloor \frac{\sqrt{6m+9}}{3} \right\rfloor$ mutually edge-disjoint rainbow spanning trees, say $T_1, T_2, \ldots, T_{\Omega_m}$, with the following properties.*

(i) Each tree has a designated distinct root.

(ii) The root of T_1 has degree $(2m-1) - 2(\Omega_m - 1)$ and has at least $(2m-1) - 4(\Omega_m - 1)$ adjacent leaves.

(iii) For $2 \leq i \leq \Omega_m$, The root of T_i has degree $(2m-1) - i - 2(\Omega_m - i)$ in and has at least $(2m-1) - 2i - 4(\Omega_m - i)$ adjacent leaves.

Before we begin the proof, we note here that Appendix A contains an example of the algorithm used in our proof to construct the edge-disjoint rainbow spanning trees that might be of use to refer to while reading the following sections.

Proof. We will use induction on the number of trees to prove this result. We can assume $m \geq 5$ since for $1 \leq m \leq 4$, $\Omega_m = 1$ and the spanning star, S_r, in which $r \in V(K_{2m})$ and r is joined to every other vertex, is clearly a rainbow spanning tree of K_{2m}. When the value of m is clear, it will cause no confusion to simply refer to Ω_m as Ω. It is worth noting that the following induction proof can be used as a recursive construction to create Ω rainbow edge-disjoint spanning trees, $T_1, T_2, ..., T_\Omega$.

For $1 \leq \psi \leq \Omega$ and rainbow edge-disjoint spanning trees, $T_1, T_2, ..., T_\psi$, let $f(\psi)$ be the proposition consisting of the three following degree and structure characteristics:

Each tree has a designated distinct root. (2.1)

The root of T_1 has degree $(2m-1)-2(\psi-1)$ and has at least $(2m-1)-4(\psi-1)$ adjacent leaves. (2.2)

For $2 \leq i \leq \psi$, The root of T_i has degree $(2m-1)-i-2(\psi-i)$ and has at least $(2m-1)-2i-4(\psi-i)$ adjacent leaves. (2.3)

In particular, note here that by (2.3), if $\psi > 1$, then the root of T_2 has degree $(2m-1)-2-2(\psi-2) = (2m-1)-2(\psi-1)$ and at least $(2m-1)-4-4(\psi-2) = (2m-1)-4(\psi-1)$ adjacent leaves, sharing these characteristics with T_1 (as stated in (2.2)).

It is useful in our construction to ensure that the rainbow edge-disjoint spanning trees have suitable characteristics that allow the properties (2.1), (2.2), and (2.3) to be established. Thus, the trees $T_1, T_2, ..., T_\Omega$ will eventually satisfy $f(\Omega)$.

We begin with some necessary notation. All vertices defined in what follows are in $V(K_{2m})$, the given edge-colored complete graph.

The proof proceeds inductively, producing a list of j edge-disjoint rainbow spanning trees from a list of $j-1$ edge-disjoint rainbow spanning trees; so for $1 \leq i \leq j \leq \Omega$, let T_i^j be the i^{th} rainbow spanning tree of the j^{th} induction step and let r_i be the designated root of T_i^j. Notice that r_i is independent of j.

Suppose T is any spanning tree of the complete graph K_{2m} with root r containing vertices y, v, w, and v', where ry and rv are distinct pendant edges in T (so y and v are leaves of T).

Then define $T' = T[r; y, v; w, v']$ to be the new graph formed from T with edges ry and rv removed and edges yw and vv' added. Formally, $T' = T[r; y, v; w, v'] = T - ry - rv + yw + vv'$. We note here that T' is also a spanning tree of K_{2m} because y and v are leaves in T, and thus adding edges yw and vv' does not create a cycle in T'.

Our inductive strategy will be to assume we have $k-1$ (where $1 < k \leq \Omega$) edge-disjoint rainbow spanning trees with suitable characteristics satisfying proposition $f(k-1)$ that yield properties (2.1), (2.2), and (2.3) with $\psi = k-1$. From those trees we will construct k edge-disjoint rainbow spanning trees with suitable characteristics that allow properties (2.1), (2.2), and (2.3) to be eventually established when $\psi = k$, thus satisfying $f(k)$.

For this construction, given any T_i^{j-1} with root r_i and distinct pendant edges $r_i y_i^j$ and $r_i v_i^j$, we define T_i^j in the following way:

$$T_i^j = T_i^{j-1}[r_i; y_i^j, v_i^j; w_i^j, v_i^{j'}] = T_i^{j-1} - r_i y_i^j - r_i v_i^j + y_i^j w_i^j + v_i^j v_i^{j'} \tag{2.4}$$

The choice of the vertices defined in (2.4) will eventually be made precise, based on the discussion which follows.

When the value of j is clear, it will cause no confusion to refer to the vertices $y_i^j, v_i^j; w_i^j, v_i^{j'}$ by omitting the superscript and instead writing $T_i^j = T_i^{j-1}[r_i; y_i, v_i; w_i, v_i']$. We now make the following remarks about the definition of T_i^j above. Recall that for $1 \leq i \leq j \leq \Omega$, r_i is independent of j, and thus is the root of both T_i^{j-1} and T_i^j. The following is easily seen to be true.

If φ is any proper edge-coloring of K_{2m} and T_i^{j-1} is a rainbow spanning tree of K_{2m} with root r_i and distinct pendant edges $r_i y_i$ and $r_i v_i$, then T_i^j as defined in (2.4) is also a rainbow spanning tree of K_{2m} if $\varphi(r_i y_i) = \varphi(v_i v_i')$ and $\varphi(r_i v_i) = \varphi(y_i w_i)$. (2.5)

Next, for $1 \leq i \leq j \leq \Omega$, let $L_i^j = \{x \mid xr_i \text{ is a pendant edge in } T_i^j\}$ (so x is a leaf adjacent to r_i in T_i^j). Define

$$L_j = \bigcap_{i=1}^{j} L_i^j. \tag{2.6}$$

Notice that if $x \in L_j$, then for $1 \leq i \leq j$, xr_i is a pendant edge in T_i^j .

We now begin our inductive proof with induction parameter k. Specifically we will prove that for $1 \leq k \leq \Omega$ there exist k edge-disjoint rainbow spanning trees, $T_1^k, T_2^k, ..., T_k^k$ satisfying $f(k)$, which for convenience we explicitly state in terms of the inductive parameter k:

1. Each tree T_i^k has a designated distinct root r_i,
2. The root of T_1^k has degree $(2m-1)-2(k-1)$ and has at least $(2m-1)-4(k-1)$ adjacent leaves,
3. For $2 \leq i \leq k$, the root of T_i^k has degree $(2m-1)-i-2(k-i)$ and has at least $(2m-1)-2i-4(k-i)$ adjacent leaves.

Base Step. The case $k = 1$ is seen to be true for all properly edge-colored complete graphs, K_{2m}, by letting r_1 be any vertex in $V(K_{2m})$ and defining $T_1^1 = S_{r_1}$, the spanning star with root r_1. It is also clear that S_{r_1} satisfies $f(1)$ since r_1 has degree $2m-1$ and has $2m-1$ adjacent leaves, as required in (2.2). Property (2.3) is vacuously true.

Induction Step. Suppose that φ is a proper edge-coloring of K_{2m} and that for some k with $1 < k \leq \Omega$, K_{2m} contains $k-1$ edge-disjoint rainbow spanning trees, $T_1^{k-1}, T_2^{k-1}, ..., T_{k-1}^{k-1}$, satisfying $f(k-1)$:

1. r_i is the root of tree T_i^{k-1} and $r_i \neq r_c$ for $1 \leq i$, $c < k$, $i \neq c$,
2. $d_{T_1^{k-1}}(r_1) = (2m-1)-2(k-2)$ and r_1 is adjacent to at least $(2m-1)-4(k-2)$ leaves in T_1^{k-1}, and

3. For $2 \leq i \leq k-1$, $d_{T_i^{k-1}}(r_i) = (2m-1) - i - 2(k-1-i)$ and r_i is adjacent to at least $(2m-1) - 2i - 4(k-1-i)$ leaves in T_i^k.

It thus remains to construct k edge-disjoint rainbow spanning trees satisfying $f(k)$.

We note here that $f(k-1)$ and the definition of L_{k-1} in (2.6) guarantee that a lower bound for $|L_{k-1}|$ can be obtained by starting with a set containing all $2m$ vertices, then removing the $k-1$ roots of $T_1^{k-1}, T_2^{k-2}, ..., T_{k-1}^{k-1}$, the (at most $4(k-2)$) vertices in $V(T_1^{k-1} \backslash \{r_1\})$ which are not leaves adjacent to r_1, and for $2 \leq i < k$, the (at most $2i + 4(k-1-i)$) vertices in $V(T_i^{k-1} \backslash \{r_i\})$ which are not leaves adjacent to r_i. Formally,

$$\begin{aligned} |L_{k-1}| &\geq 2m - (k-1) - 4(k-2) - \sum_{i=2}^{k-1}(2i + 4(k-1-i)) \\ &= 2m - (k-1) - 4(k-2) - (3k^2 - 11k + 10) \\ &= 2m - 3k^2 + 6k - 1. \end{aligned} \tag{2.7}$$

Knowing $|L_{k-1}|$ is useful because later we will show that if $|L_{k-1}| \geq 6k - 7$, then from $T_1^{k-1}, T_2^{k-1}, ..., T_{k-1}^{k-1}$ we can construct k rainbow edge-disjoint spanning trees which satisfy proposition $f(k)$. As the reader might expect, it is from here that the bound on Ω is obtained: it actually follows that since $k \leq \Omega$, $|L_{k-1}| \geq 6k - 7$.

First select any two distinct vertices $r_k, w_k^k \in L_{k-1}$; since it will cause no confusion, we will write w_k for w_k^k. Set r_k equal to the root of the k^{th} tree, T_k^k. Later, $r_k w_k$ will be an edge removed from T_k^k. For now, the two special vertices r_k and w_k play a role in the construction of T_i^k from T_i^{k-1} for $1 \leq i < k$. For convenience, we explicitly state and observe the following

$$\begin{aligned} &\text{Since } r_k \text{ and } w_k \text{ are distinct vertices in } L_{k-1} \text{ (defined in (2.6)),} \\ &r_k \text{ and } w_k \text{ are leaves adjacent to } r_i \text{ for } 1 \leq i < k. \end{aligned} \tag{2.8}$$

For the sake of clarity, having selected r_k and w_k, we now discuss how to construct the trees $T_1^k, T_2^k, ..., T_{k-1}^k$ before returning to our discussion of the construction of T_k^k (though in actuality T_k^k is formed recursively as we are constructing $T_1^k, T_2^k, ..., T_{k-1}^k$).

For $1 \leq i < k$, we will find suitable vertices v_i^k, w_i^k, and $v_i^{k'}$, which for convenience we refer to as v_i, w_i, and v_i' respectively, and define T_i^k in the following way:

$$\begin{aligned} &T_i^k = T_i^{k-1}[r_i; r_k, v_i; w_i, v_i'] \\ &\text{where } \varphi(r_i r_k) = \varphi(v_i v_i') \text{ and } \varphi(r_i v_i) = \varphi(r_k w_i) \end{aligned} \tag{2.9}$$

It is clear by (2.5) that for $1 \leq i < k$, since T_i^{k-1} is a rainbow spanning tree of K_{2m}, if v_i is chosen so that $v_i r_i$ is a pendant edge in T_i^{k-1} with $v_i \neq r_k$, then T_i^k is also a rainbow spanning tree of K_{2m} (recall from (2.8) that $r_k \in L_{k-1}$, so by (2.6) $r_k r_i$ is a pendant edge in T_i^{k-1}).

Further, since $r_k, w_k \in L_{k-1}$, it is clear from (2.9) that (1) $r_k, v_i \notin L_k$, and (2) all leaves adjacent to r_i in T_i^k are leaves adjacent to r_i in T_i^{k-1}. Therefore $|L_k| < |L_{k-1}|$. (2.10)

Lastly, since the trees $T_1^{k-1}, T_2^{k-1}, ..., T_{k-1}^{k-1}$ satisfy $f(k-1)$, it can be seen that $T_1^k, T_2^k, ...,$ T_{k-1}^k satisfy $f(k)$, as the following shows.

First, clearly (2.1) is satisfied. Further, for $1 \leq i < k$, when T_i^k is formed from T_i^{k-1} (see (2.9)), it can easily be seen that the degree of r_i is decreased by 2 and the number of leaves adjacent to r_i is decreased by at most 4.

(i.) T_1^k

By our induction hypothesis, we have that $d_{T_1^{k-1}}(r_1) = (2m-1) - 2(k-2)$ and that r_1 is adjacent to at least $(2m-1) - 4(k-2)$ leaves in T_1^{k-1}. From (2.9) we have that $d_{T_1^k}(r_1) = d_{T_1^{k-1}}(r_1) - 2 = (2m-1) - 2(k-2) - 2 = (2m-1) - 2(k-1)$ and that r_1 is adjacent to at least $(2m-1) - 4(k-2) - 4 = (2m-1) - 4(k-1)$ leaves in T_1^k. So (2.2) of $f(k)$ is satisfied.

(ii.) T_i^k, $2 \le i < k$

By our induction hypothesis, we have that $d_{T_i^{k-1}}(r_i) = (2m-1) - i - 2(k-1-i)$ and that r_i is adjacent to at least $(2m-1) - 2i - 4(k-1-i)$ leaves in T_i^k. From (2.9) we have that $d_{T_i^k}(r_i) = d_{T_i^{k-1}}(r_i) - 2 = (2m-1) - i - 2(k-1-i) - 2 = (2m-1) - i - 2(k-i)$ and that r_i is adjacent to at least $(2m-1) - 2i - 4(k-1-i) - 4 = (2m-1) - 2i - 4(k-i)$ leaves in T_i^k. So (2.3) of $f(k)$ is satisfied except possibly when $i = k$.

Lastly, we can observe that once v_i is selected, vertices w_i and v_i' are determined by the required property from (2.9) that $\varphi(r_i r_k) = \varphi(v_i v_i')$ and $\varphi(r_i v_i) = \varphi(r_k w_i)$.

It remains to ensure that the trees, $T_1^k, T_2^k, ..., T_{k-1}^k$, are all edge-disjoint. This is also proved using the induction hypothesis that $T_1^{k-1}, T_2^{k-1}, ..., T_{k-1}^{k-1}$ are all edge-disjoint, which allows us to show that $T_1^k, T_2^k, ..., T_{k-1}^k$ are all edge-disjoint.

Now, while forming the rainbow edge-disjoint spanning trees, $T_1^k, T_2^k, ..., T_{k-1}^k$, we simultaneously construct the k^{th} rainbow spanning tree, T_k^k, from a sequence of inductively defined graphs, $T_k^k(1), T_k^k(2), ..., T_k^k(k) = T_k^k$ where at the i^{th} induction step, the formation of $T_k^k(i)$ depends on the choice of v_i used in the construction of T_i^k: for $2 \le i \le k$ define

$$\begin{aligned} &T_k^k(i) = S_{r_k} - r_k w_1 - ... - r_k w_i + w_1 w_1' + ... + w_i w_i', \\ &\text{where } \varphi(w_1 w_1') = \varphi(r_k w_k) \text{ and } \varphi(w_i w_i') = \varphi(r_k w_{i-1}) \text{ for } 2 \le i \le k. \end{aligned} \tag{2.11}$$

Note that for $1 \le i \le k-1$, the choice of v_i determines $T_k^k(i)$; the formation of $T_k^k(k)$ is dictated by $T_k^k(k-1)$ since w_k' is determined by requiring that $\varphi(w_k w_k') = \varphi(r_k w_{k-1})$. It is worth explicitly stating that

$$\begin{aligned} &T_k^k = T_k^k(k) = S_{r_k} - r_k w_1 - ... - r_k w_k + w_1 w_1' + ... + w_k w_k', \\ &\text{where } \varphi(w_1 w_1') = \varphi(r_k w_k) \text{ and } \varphi(w_c w_c') = \varphi(r_k w_{c-1}) \text{ for } 2 \le c \le k \end{aligned} \tag{2.12}$$

Observe that T_k^k is a rainbow graph since each edge removed from S_{r_k} is replaced by a corresponding edge of the same color. Also, one can easily see that: T_k^k has $2m-1$ edges;

$d_{T_k^k}(r_k) = (2m-1) - k$ since $r_k \notin \{w_1', w_2', ..., w_k'\}$; and r_k has at least $(2m-1) - 2k$ adjacent leaves. Therefore, condition (2.3) of $f(k)$ is satisfied. So it remains to show that T_k^k is acyclic and contains no edges in the trees T_i^k for $1 \leq i \leq k-1$.

Finally, we have noted previously, but restate here because of its importance,

$$\text{For } 1 \leq i < k, \text{ once } v_i \text{ is chosen, } T_i^k \text{ and } T_k^k(i) \text{ are completely determined by the constructions described in (2.9) and (2.11) respectively.} \tag{2.13}$$

Due to the fact highlighted above in (2.13), our strategy will be to select a suitable v_i and construct T_i^k from T_i^{k-1}, while simultaneously constructing $T_k^k(i)$ from $T_k^k(i-1)$. In doing so, we restrict the choices for each v_i in order to achieve the following three properties:

(C1) The edges in $T_a^k, 1 \leq a < i$ do not appear in T_i^k,

(C2) The edges in T_k^k do not appear in T_i^k, $1 \leq i < k$, and

(C3) T_k^k is acyclic

To that end, we let

$$L_{k-1}^* = L_{k-1} \backslash \{r_k, w_k\} \tag{2.14}$$

and let v_i be any vertex for which the following properties are satisfied (so by (2.13), this choice completes the formation of T_i^k and $T_k^k(i)$ for $1 \leq i < k$):

(R1) $v_i \in L_{k-1}^*$,

(R2) For $1 \leq c < k$, $c \neq i$, $\varphi(v_i r_c) \neq \varphi(r_i r_k)$,

(R3) For $1 \leq a < i$, $\varphi(v_i r_i) \neq \varphi(r_a v_a)$,

(R4) For $i < b < k$, $\varphi(v_i r_i) \neq \varphi(r_k r_b)$,

(R5) $\varphi(v_i r_i) \neq \varphi(r_k w_k)$,

(R6) For $1 \leq a < i$, $\varphi(v_i r_i) \neq \varphi(r_k w_a')$, (R7) For $2 \leq i < k$, $\varphi(v_i r_i) \neq \varphi(r_k \alpha)$,

where α is the vertex such that $\varphi(w_k \alpha) = \varphi(r_k w_{i-1})$,

(R8) For $i = 1$ and for $1 \leq c < k$, $\varphi(v_1 r_1) \neq \varphi(r_k \alpha)$,

for each vertex α incident with the edge of color $\varphi(r_k w_k)$ in T_c^{k-1},

(R9) For $2 \leq i < k$, $1 \leq a < i$, and for $i \leq b < k$, $\varphi(v_i r_i) \neq \varphi(r_k \alpha)$,

for each vertex α incident with the edge of color $\varphi(r_k w_{i-1})$ in T_a^k and in T_b^{k-1},

(R10) For $1 \leq i < k$, $\varphi(v_i w_k) \neq \varphi(r_i r_k)$,

(R11) For $1 \leq d \leq k-2$, $\varphi(v_{k-1} r_{k-1}) \neq \varphi(w_k r_d)$.

From the observation in (2.7), we know that $\left|L_{k-1}^*\right| \geq 2m - 3k^2 + 6k - 3$.

An upper bound for the number of vertices eliminated through items (R2 - R11) as candidates for v_i is achieved when $i = k-1$. In this case, the number of vertices eliminated by R2, R3, ..., R11 is $(k-2), (k-2), 0, 1, (k-2), 1, 0, 2(k-1), 1, (k-2)$ respectively, the sum of which is $6k - 7$. Now, since the induction hypothesis includes the condition $k \leq \Omega$, we can observe the following.

First, from $f(\Omega)$ and the definition of $L_{\Omega-1}$, we can follow the same steps as we did in (2.7) to see that $|L_{\Omega-1}| \geq 2m - 3\Omega^2 + 6\Omega - 1$ and further, that $\left|L_{\Omega-1}^*\right| \geq 2m - 3\Omega^2 + 6\Omega - 3$. Now, since by the induction hypothesis $k \leq \Omega$ and by (2.10) and (2.14) $\left|L_{i-1}^*\right| > |L_i^*|$ for $2 \leq i \leq k-1$, we have the following:

$$
\begin{aligned}
\left|L^*_{k-1}\right| &\geq \left|L^*_{\Omega-1}\right| \\
&\geq 2m - 3\Omega^2 + 6\Omega - 3 \\
&= 2m - 3(\left\lfloor \frac{\sqrt{6m+9}}{3} \right\rfloor)^2 + 6\left\lfloor \frac{\sqrt{6m+9}}{3} \right\rfloor - 3 \\
&\geq 2m - (2m+3) + 2\sqrt{6m+9} - 3 \\
&= 2\sqrt{6m+9} - 6 \qquad (2.15)\\
&= \frac{6}{3}\sqrt{6m+9} - 6 \\
&\geq 6\Omega - 6 \\
&> 6\Omega - 7 \\
&\geq 6k - 7.
\end{aligned}
$$

In summary, we have that $\left|L^*_{k-1}\right| \geq \left|L^*_{\Omega-1}\right| > 6\Omega - 7 \geq 6k - 7$. Therefore, $\left|L^*_{k-1}\right| > 6k - 7$, and so such a vertex v_i meeting the restrictions in (R1 - R11) exists. The following cases show that this choice of v_i ensures that (C1), (C2), and (C3) hold.

2.1 Case 1 (C1): Edges in $T^k_a, 1 \leq a < i$ do not appear in T^k_i

First, by the induction hypothesis we know that the trees $T^{k-1}_1, T^{k-1}_2, ..., T^{k-1}_{k-1}$ are all rainbow edge-disjoint and spanning. Inductively, we also assume for some i with $2 \leq i < k$ the trees $T^k_1, T^k_2, ..., T^k_{i-1}$ are edge-disjoint rainbow spanning trees as well. By (2.9), regardless of the choice of v_i, the only edges in T^k_i $(1 \leq i < k)$ that are not in T^{k-1}_i are $v_iv'_i$ and r_kw_i. Thus, if we can prove that the edges in $(E(T^{k-1}_i)\backslash\{r_iv_i, r_ir_k\}) \cup \{v_iv'_i, r_kw_i\}$ are not in T^k_a, $1 \leq a < i$, we will have shown that the trees $T^k_1, T^k_2, ..., T^k_i$ are all edge-disjoint rainbow and spanning; so by induction, $T^k_1, T^k_2, ..., T^k_{k-1}$ are edge-disjoint rainbow spanning trees.

To that end, for the remainder of Case 1 suppose that $2 \leq i < k$, $1 \leq a < i$, and $i < b < k$ and define the following sets of edges.

1. $E_{old}(T_a^k) = \{xy \mid xy \in E(T_a^{k-1}) \cap E(T_a^k)\}$

2. $E_{new}(T_a^k) = E(T_a^k) \backslash E(T_a^{k-1}) = \{v_a v_a', \, r_k w_a\}$

3. $E_{old}(T_i^k) = \{xy \mid xy \in E(T_i^{k-1}) \cap E(T_i^k)\}$

4. $E_{new}(T_i^k) = E(T_i^k) \backslash E(T_i^{k-1}) = \{v_i v_i', \, r_k w_i\}$

Observe that by (2.9), $E_{old}(T_a^k) \cap E_{new}(T_a^k) = \emptyset$ and $E(T_a^k) = E_{old}(T_a^k) \cup E_{new}(T_a^k)$. Similarly, $E_{old}(T_i^k) \cap E_{new}(T_i^k) = \emptyset$ and $E(T_i^k) = E_{old}(T_i^k) \cup E_{new}(T_i^k)$.

Since the trees $T_1^k, T_2^k, ..., T_{k-1}^k$ are formed sequentially, it is clearly necessary to prohibit edges $v_i v_i'$ and $r_k w_i$ from appearing in T_a^k. It is also very useful to prohibit edges $v_i v_i'$ and $r_k w_i$ from appearing in T_b^{k-1}.

Consequently, when v_i was selected to satisfy (R1 - R11) it was done in such a way that ensures the following six properties are satisfied:

(P1) $v_i v_i', \, r_k w_i \notin E_{old}(T_a^k)$,

(P2) $v_i v_i', \, r_k w_i \notin E_{new}(T_a^k)$,

(P3) $v_i v_i', \, r_k w_i \notin E(T_b^{k-1})$,

(P4) $E_{old}(T_i^k) \cap E_{old}(T_a^k) = \emptyset$,

(P5) $E_{old}(T_i^k) \cap E_{new}(T_a^k) = \emptyset$,

(P6) $E_{old}(T_i^k) \cap E(T_b^{k-1}) = \emptyset$.

It is clear that if properties (P1 - P6) are satisfied, then T_i^k is edge-disjoint from the trees, T_a^k and T_b^{k-1}. We consider edges $v_i v_i'$ and $r_k w_i$ in turn for properties (P1 - P3), then address properties (P4 - P6).

2.1.1 Property (P1) for v_iv_i'

Since $E_{old}(T_a^k) \subset E(T_a^{k-1})$, we can prove v_iv_i' is not an edge in $E_{old}(T_a^k)$ by showing that $v_iv_i' \notin E(T_a^{k-1})$.

Recall from (R1) and (2.14) that because $v_i \in L_{k-1}^*$, v_i is a leaf adjacent to the root r_c in T_c^{k-1}, for $1 \leq c < k$. Therefore, to show that $v_iv_i' \notin E(T_a^{k-1})$, we need only prove that $v_i' \neq r_a$. The following argument shows that (R2) guarantees this property.

Suppose to the contrary that $v_i' = r_a$. Then $v_iv_i' = v_ir_a$ and by (2.9), $\varphi(v_ir_a) = \varphi(v_iv_i') = \varphi(r_ir_k)$, contradicting (R2). It follows that $v_i' \neq r_a$ so $v_iv_i' \notin E_{old}(T_a^k)$, as required.

2.1.2 Property (P2) for v_iv_i'

Recall that $E_{new}(T_a^k) = \{v_av_a', r_kw_a\}$. Thus, to prove that $v_iv_i' \notin E_{new}(T_a^k)$ for $1 \leq a < i$, we need only show that $v_iv_i' \neq v_av_a'$ and $v_iv_i' \neq r_kw_a$. We consider each in turn.

(i.) $v_iv_i' \neq v_av_a'$

By (2.9), we have that $\varphi(v_iv_i') = \varphi(r_ir_k)$ and $\varphi(v_av_a') = \varphi(r_ar_k)$. But, by property (1) of $f(\psi)$ when $\psi = k-1$ we know $r_i \neq r_a$ and so $\varphi(r_ir_k) \neq \varphi(r_ar_k)$. It follows that $\varphi(v_iv_i') \neq \varphi(v_av_a')$ and, therefore, $v_iv_i' \neq v_av_a'$.

(ii.) $v_iv_i' \neq r_kw_a$

Assume that $v_iv_i' = r_kw_a$ and recall from (2.14) that because $v_i \in L_{k-1}^*$, $v_i \neq r_k$. Therefore, $v_i = w_a$. By (2.9), $\varphi(v_iv_i') = \varphi(r_kr_i)$, so since we are assuming that $v_iv_i' = r_kw_a$, clearly $\varphi(r_kr_i) = \varphi(r_kw_a)$ and so $w_a = r_i = v_i$. But because $v_i \in L_{k-1}^*$, $v_i \neq r_i$ and this is a contradiction.

Combining the above two arguments, it is clear that $v_iv_i' \notin E_{new}(T_a^k)$, as required.

2.1.3 Property (P3) for v_iv_i'

Recall from (2.14) that $v_i \in L_{k-1}^*$, so r_bv_i is a pendant edge with leaf v_i in T_b^{k-1}, for $i < b < k$. Thus, v_iv_i' would only be an edge in T_b^{k-1} if $v_i' = r_b$. As in Section 2.1.1 above, (R2) prevents v_i' from equalling r_b by guaranteeing that $\varphi(v_i r_b) \neq \varphi(r_i r_k)$ and therefore, $v_i v_i' \notin E(T_b^{k-1})$, as required.

2.1.4 Property (P1) for $r_k w_i$

Recall from (2.6) that $r_k \in L_{k-1}$, so $r_k r_a$ is a pendant edge in T_a^{k-1} with leaf r_k. Therefore, from (2.9) it is clear that $r_k r_a \notin E(T_a^k)$ since it is removed from T_a^{k-1} in forming T_a^k. So r_k is not incident with any edges in $E_{old}(T_a^k)$ and thus, $r_k w_i$ cannot be an edge in $E_{old}(T_a^k)$, as required.

2.1.5 Property (P2) for $r_k w_i$

Recall that $E_{new}(T_a^k) = \{v_a v_a', r_k w_a\}$. To show that $r_k w_k \notin E_{new}(T_a^k)$, we prove that $r_k w_i \neq r_k w_a$ and $r_k w_i \neq v_a v_a'$ for $1 \leq a < i$. We consider each in turn.

(i.) $r_k w_i \neq r_k w_a$

To show that $r_k w_i \neq r_k w_a$, we need only show that $w_i \neq w_a$.

By (2.9) we have that $\varphi(r_k w_i) = \varphi(r_i v_i)$ and $\varphi(r_k w_a) = \varphi(r_a v_a)$. So if $r_k w_i = r_k w_a$, then $\varphi(v_i r_i) = \varphi(r_a v_a)$, contradicting (R3). Therefore, $r_k w_i \neq r_k w_a$, as required.

(ii.) $r_k w_i \neq v_a v_a'$

Assume that $r_k w_i = v_a v_a'$. Recall from (2.14) that because $v_a \in L_{k-1}^*$, $v_a \neq r_k$. Therefore, $v_a = w_i$. By (2.9), $\varphi(v_a v_a') = \varphi(r_a r_k)$, so since we are assuming that $r_k w_i = v_a v_a'$, then $\varphi(r_k w_i) = \varphi(r_k r_a)$ and it follows that $r_a = w_i = v_a$. But this is a contradiction because $v_a \in L_{k-1}^*$ so by (2.14), $v_a \neq r_a$.

Combining the above two arguments, it is clear that $r_k w_i \notin E_{new}(T_a^k)$, as required.

2.1.6 Property (P3) for $r_k w_i$

Recall that by (2.8), because r_k was chosen to be in L_{k-1}, r_k is a leaf adjacent to the root of T_b^{k-1}, $i < b < k$. Thus, to show $r_k w_i \notin E(T_b^{k-1})$, we need only prove that $w_i \neq r_b$.

By (2.9), we have that $\varphi(r_k w_i) = \varphi(v_i r_i)$. So if $w_i = r_b$, then $r_k w_i = r_k r_b$ and $\varphi(v_i r_i) = \varphi(r_k r_b)$, contradicting (R4). Therefore, $r_k w_i \notin E(T_b^{k-1})$, as required.

2.1.7 Properties (P4), (P5), and (P6)

We consider each property, (P4), (P5), and (P6), in turn.

(i.) Property (P4)

By our induction hypothesis, the trees, $T_1^{k-1}, T_2^{k-1}, ..., T_{k-1}^{k-1}$ are all edge disjoint. So (P4) follows because $E_{old}(T_i^k) \subset E(T_i^{k-1})$ and $E_{old}(T_a^k) \subset E(T_a^{k-1})$.

(ii.) Property (P5)

Since $a < i$, from (P3) (replacing i with a), it follows that $\{v_a v'_a, r_k w_a\} \cap E(T_c^{k-1}) = \emptyset$, for $a < c < k$. In particular, since $i > a$, it follows that $E_{new}(T_a^k) \cap E(T_i^{k-1}) = \emptyset$. And lastly, since $E_{old}(T_i^k) \subset E(T_i^{k-1})$, we have that $E_{old}(T_i^k) \cap E_{new}(T_a^k) = \emptyset$.

(iii.) Property (P6)

Again, by our induction hypothesis, the trees, $T_1^{k-1}, T_2^{k-1}, ..., T_{k-1}^{k-1}$ are all edge-disjoint. It follows that $E_{old}(T_i^k) \cap E(T_b^{k-1}) = \emptyset$ because $E_{old}(T_i^k) \subset E(T_i^{k-1})$.

Therefore, properties (P4 - P6) hold for $E_{old}(T_i^k)$.

The above Sections 2.1.1 – 2.1.7 ensure that properties (P1 - P6) hold. As stated above, since these six properties hold, the trees $T_1^k, T_2^k, ..., T_{k-1}^k$ are all edge-disjoint and further, from (2.9), are also rainbow and spanning.

2.2 Case 2 (C2): Edges in T_k^k do not appear in T_i^k

Recall from (2.11) that T_k^k is defined by a sequence, $T_k^k(1), T_k^k(2), ..., T_k^k(k)$, and from (2.13) that at the i^{th} induction step, $T_k^k(i)$ was determined by the choice of v_i. It is convenient to restate (2.11) and (2.12) here:

$$T_k^k(i) = S_{r_k} - r_k w_1 - ... - r_k w_i + w_1 w_1' + ... + w_i w_i',$$

where $\varphi(w_1 w_1') = \varphi(r_k w_k)$ and $\varphi(w_i w_i') = \varphi(r_k w_{i-1})$ for $2 \leq i \leq k$.

$$T_k^k = S_{r_k} - r_k w_1 - ... - r_k w_k + w_1 w_1' + ... + w_k w_k',$$

where $\varphi(w_1 w_1') = \varphi(r_k w_k)$ and for $2 \leq c \leq k$, $\varphi(w_c w_c') = \varphi(r_k w_{c-1})$,

For the remainder of Case 2, suppose that $1 \leq i < k$, $1 \leq a < i$, and $i < b < k$.

In order to prevent edges in T_k^k from also appearing in T_i^k, we will now show that T_i^k has been constructed in such a way that $T_k^k(i)$ and T_k^k satisfy the following properties:

(P7) $E(T_k^k(i)) \cap E(T_a^k) = \emptyset$

(P8) $E(T_k^k(i)) \cap E(T_b^{k-1}) = \{r_k r_b\}$

(P9) $E(T_k^k(i)) \cap E_{old}(T_i^k) = \emptyset$

(P10) $E(T_k^k(i)) \cap E_{new}(T_i^k) = \emptyset$

(P11) $w_k w_k' \notin E(T_i^k)$

We note here that by (2.9), when T_b^k was constructed from T_b^{k-1}, edge $r_k r_b$ was removed, so it does not appear in T_b^k. Therefore, it is not necessary to prevent $r_k r_b$ from being an edge in $T_k^k(i)$ nor T_k^k.

Proving the above five properties will be done inductively. We show in the base step that $T_k^k(1)$ satisfies properties (P7 - P10) with $i = 1$, and then show that for $2 \leq i < k$, $T_k^k(i)$ satisfies the same four properties before finally proving property (P11).

The following preliminary result will be useful in proving properties (P7 - P11).

2.2.1 Preliminary Result: $w_i \neq w_k$

Recall from (2.8) that $w_k \in L_{k-1}$ was selected with r_k before any of the rainbow spanning trees $T_1^{k-1}, T_2^{k-1}, ..., T_{k-1}^{k-1}$ were revised. It will be useful to show that the vertices $w_i \in T_i^k$, $1 \leq i < k$, cannot equal w_k.

From (2.9), we have that $\varphi(v_i r_i) = \varphi(r_k w_i)$. So if $w_i = w_k$, then $\varphi(v_i r_i) = \varphi(r_k w_k)$ contradicting (R5). Therefore, $w_i \neq w_k$.

2.2.2 Base Step: $i = 1$

Observe that for $2 \leq b < k$, $E(S_{r_k}) \cap E(T_b^{k-1}) = \{r_k r_b\}$ and $E(S_{r_k}) \cap E_{old}(T_1^k) = \emptyset$ since by (2.9), $r_k r_1$ is removed from T_1^{k-1} when forming T_1^k. Further, it is clear from (2.11) that the only edge in $T_k^k(1)$ that is not in S_{r_k} is $w_1 w_1'$.

(i.) (P7)

Since $i = 1$, there do not exist any such trees T_a^k since $1 \leq a < i$ and so property (P7) is vacuously true.

(ii.) (P8) and (P9)

First, recall that $E_{old}(T_1^k) \subset E(T_1^{k-1})$. To establish properties (P8) and (P9), we show that $w_1 w_1' \notin E(T_c^{k-1})$ for $1 \leq c < k$.

Suppose to the contrary that $w_1 w_1' \in E(T_c^{k-1})$. Recall from (2.11) that $\varphi(w_1 w_1') = \varphi(r_k w_k)$. So if $w_1 w_1' \in E(T_c^{k-1})$, then w_1 is a vertex incident to the edge of color $\varphi(r_k w_k)$ in T_c^{k-1}. But this is impossible since from (2.9) we have that $\varphi(v_1 r_1) = \varphi(r_k w_1)$ and from (R8) that $\varphi(v_1 r_1) \neq \varphi(r_k \alpha)$, where α is a vertex incident to the edge of color $\varphi(r_k w_k)$ in T_c^{k-1}. Therefore, $w_1 w_1' \notin E(T_c^{k-1})$ and $T_k^k(1)$ satisfies properties (P8) and (P9).

(iii.) (P10)

Recall that $E_{new}(T_i^k) = \{v_i v_i', r_k w_i\}$. To establish (P10) for $T_k^k(1)$, we need only show that $w_1 w_1' \neq v_1 v_1'$ and $w_1 w_1' \neq r_k w_1$. We consider each in turn.

(a.) $w_1 w_1' \neq v_1 v_1'$

Recall from (2.9) that $\varphi(v_1 v_1') = \varphi(r_k r_1)$ and from (2.11) that $\varphi(w_1 w_1') = \varphi(r_k w_k)$. So if $w_1 w_1' = v_1 v_1'$, then $\varphi(r_k w_k) = \varphi(r_k r_1)$ and so $w_k = r_1$. But this is not possible because by (2.8) $w_k \in L_{k-1}$ and so $w_k \neq r_1$. Therefore, $w_1 w_1' \neq v_1 v_1'$.

(b.) $w_1 w_1' \neq r_k w_1$

Recall from (2.11) that $\varphi(w_1 w_1') = \varphi(r_k w_k)$. So if $w_1 w_1' = r_k w_1$, then $\varphi(r_k w_k) = \varphi(r_k w_1)$ and so $w_k = w_1$, contradicting the result in Section 2.2.1. Thus, $w_1 w_1' \neq r_k w_1$.

Therefore, property (P10) holds for $T_k^k(1)$ and we have established our base step.

2.2.3 Property (P7) for $2 \leq i < k$

From (2.11), it is clear that the only edge in $T_k^k(i)$ that differs from $T_k^k(i-1)$ is $w_i w_i'$. Therefore, since by induction we have that $T_k^k(i-1)$ satisfies (P7), in order to prove property (P7) is satisfied for $T_k^k(i)$, we need only show that $w_i w_i'$ is not an edge in T_a^k, $1 \leq a < i$.

To that end, suppose to the contrary that $w_i w_i' \in E(T_a^k)$. Recall from (2.11) that $\varphi(w_i w_i') = \varphi(r_k w_{i-1})$. So if $w_i w_i' \in E(T_a^k)$, then w_i is a vertex incident to the edge of color $\varphi(r_k w_{i-1})$ in T_a^k. But this is impossible since from (2.9) we have that $\varphi(v_i r_i) = \varphi(r_k w_i)$ and from (R9) that $\varphi(v_i r_i) \neq \varphi(r_k \alpha)$, where α is a vertex incident to the edge of color $\varphi(r_k w_{i-1})$ in T_a^k. Therefore, $w_i w_i' \notin E(T_a^k)$ and $T_k^k(i)$ satisfies property (P7).

2.2.4 Properties (P8) and (P9) for $2 \leq i < k$

Observe again that $E_{old}(T_i^k) \subset E(T_i^{k-1})$. As in Section 2.2.3, to prove properties (P8) and (P9) for $T_k^k(i)$, we can show that $w_i w_i' \notin E(T_d^{k-1})$, $i \leq d < k$.

For $i \leq d < k$, property (R9), which guarantees $\varphi(v_i r_i) \neq \varphi(r_k \alpha)$, where α is a vertex incident to the edge of color $\varphi(r_k w_{i-1})$ in T_d^{k-1}, ensures $w_i w_i' \notin E(T_d^{k-1})$, thus ensuring that (P8) and (P9) hold for $T_k^k(i)$. The argument has been omitted here due to its similarity to the argument used above for (P7) in Section 2.2.3.

2.2.5 Property (P10) for $2 \leq i < k$

To prove (P10) for $T_k^k(i)$, we need only show that $w_i w_i' \neq v_i v_i'$ and $w_i w_i' \neq r_k w_i$. We consider each in turn.

(i.) $w_i w_i' \neq v_i v_i'$

Recall from (2.9) that $\varphi(v_i v_i') = \varphi(r_k r_i)$ and from (2.11) that $\varphi(w_i w_i') = \varphi(r_k w_{i-1})$. If $w_i w_i' = v_i v_i'$, then $\varphi(r_k w_{i-1}) = \varphi(r_k r_i)$ and so $w_{i-1} = r_i$. But $r_k r_i \in E(T_i^{k-1})$ and $r_k w_{i-1} \in E(T_{i-1}^k)$; so if $w_{i-1} = r_i$, this contradicts property (P3) in the $i-1^{th}$ induction step, which in particular (i.e. when $b = i$) ensures that $r_k w_{i-1} \notin E(T_i^{k-1})$. Therefore, $w_i w_i' \neq v_i v_i'$, as required.

(ii.) $w_i w_i' \neq r_k w_i$

Recall from (2.11) that $\varphi(w_i w_i') = \varphi(r_k w_{i-1})$. If $w_i w_i' = r_k w_i$, then $\varphi(r_k w_{i-1}) = \varphi(r_k w_i)$ and so $w_{i-1} = w_i$. However, this is impossible by the result in Section 2.1.5 which, in particular, proved that $r_k w_i \neq r_k w_a$ for $1 \leq a < i$. Thus, $w_i w_i' \neq r_k w_i$.

Therefore, property (P10) holds for $T_k^k(i)$, as required.

2.2.6 Property (P11) for $w_k w_k'$

The above sections of Case 2 ensure that the rainbow spanning trees $T_1^k, T_2^k, ..., T_{k-1}^k$ and the rainbow spanning graph, $T_k^k(k-1)$ are all edge-disjoint. Thus, it remains to show that $T_1^k, T_2^k, ..., T_{k-1}^k$ and T_k^k are all edge-disjoint. As above, recall from (2.11) that the only edge in T_k^k that differs from $T_k^k(k-1)$ is $w_k w_k'$. Therefore, showing property (P11) holds will prove that $T_1^k, T_2^k, ...T_{k-1}^k$ and T_k^k are edge-disjoint.

First, observe from (2.8) that since $w_k \in L_{k-1}$, w_k is a leaf adjacent to the root r_i in T_i^{k-1} for $1 \leq i < k$. So if $w_k w_k' \in E(T_i^k)$, $w_k w_k' = w_i r_k$, $v_i v_i'$, or $w_k r_i$. We consider each in turn.

(i.) $w_k w_k' \neq w_i r_k$

From (2.8) we know that $w_k \neq r_k$. So if $w_k w_k' = w_i r_k$, then $w_k = w_i$, contradicting the preliminary result in Section 2.2.1. Therefore, $w_k w_k' \neq w_i r_k$, as required.

(ii.) $w_k w_k' \neq v_i v_i'$

Recall from (2.14) that since $v_i \in L_{k-1}^*$, $v_i \neq w_k$. So if $w_k w_k' = v_i v_i'$, then $w_k = v_i'$. From (2.9) we know that $\varphi(v_i v_i') = \varphi(r_i r_k)$, so if $w_k = v_i'$, then $\varphi(v_i w_k) = \varphi(r_i r_k)$, contradicting (R10). Therefore, $w_k w_k' \neq v_i v_i'$, as required.

(iii.) $w_k w_k' \neq w_k r_i$

Recall from (2.11) that $\varphi(w_k w_k') = \varphi(r_k w_{k-1})$ and suppose that $w_k w_k' = w_k r_i$. First observe that $i \neq k-1$ since $r_k w_{k-1} \in E(T_{k-1}^k)$ and we know from (2.8) and Section 2.2.1 that $w_k \neq r_k$ and $w_k \neq w_{k-1}$.

Now, for $1 \leq i \leq k-2$, if $w_k w_k' = w_k r_i$ then $r_i = w_k'$. But from (2.9) and (2.11) if $r_i = w_k'$ then $\varphi(w_k w_k') = \varphi(r_k w_{k-1}) = \varphi(v_{k-1} r_{k-1}) = \varphi(w_k r_i)$, contradicting (R11). Therefore, $w_k w_k' \neq w_k r_i$, as required.

It follows that $w_k w_k' \notin E(T_i^k)$, $1 \leq i < k$.

The above Sections 2.2.1 - 2.2.6 ensure that the trees $T_1^k, T_2^k, ..., T_{k-1}^k$ and the graph T_k^k are all edge-disjoint. Further, from (2.9) it is clear that $T_1^k, T_2^k, ..., T_{k-1}^k$ are all rainbow spanning trees and from (2.12) that T_k^k is a spanning rainbow graph (since for every leaf, w_c, $1 \leq c \leq k$, which is adjacent to r_k and for which $r_k w_c$ is removed from T_k^k, there exists w_c' such that the edge $w_c w_c'$ is added to T_k^k and edge $w_d w_d'$ in T_k^k such that $\varphi(w_d w_d') = \varphi(r_k w_c)$, where $d \equiv c + 1 \mod k$.)

2.3 Case 3 (C3): Preventing cycles from appearing in T_k^k

Properties (C1) and (C2) in the previous sections guarantee that the rainbow spanning trees $T_1^k, T_2^k, ..., T_{k-1}^k$ and the rainbow spanning graph T_k^k are all edge-disjoint. Thus, it remains to prove that T_k^k is acyclic and, therefore, a tree. This is proved inductively, showing that for $1 \leq i \leq k$, $T_k^k(i)$ is acyclic. Formally, we will show the following two properties:

(P12) $T_k^k(i)$ is acyclic for $1 \leq i < k$, and

(P13) T_k^k is acyclic

We consider each in turn.

2.3.1 Property (P12)

Proving $T_k^k(i)$ is acyclic will also be done inductively. For our base step, we let $T_k^k(0) = S_{r_k}$ and observe that this graph is clearly acyclic.

It is clear from (2.11) that for $1 \leq i < k$, $T_k^k(i) = T_k^k(i-1) - r_k w_i + w_i w_i'$. Therefore, since by induction we have that $T_k^k(i-1)$ satisfies (P12), in order to prove $T_k^k(i)$ is acyclic, we need only show that adding $w_i w_i'$ to $T_k^k(i-1) - r_k w_i$ does not create a cycle. Let $T_k^k(i-1)^* = T_k^k(i-1) - r_k w_i$.

Now, from (2.11) observe that all of the edges in $T_k^k(i-1)$ are of the form $r_k x$, $r_k w_a'$, and $w_a w_a'$, where $1 \leq a < i$ and $x \in V(K_{2m}) \backslash (\{\bigcup_{a=1}^{i-1} w_a, w_a'\} \cup \{r_k\})$. Thus, $w_i \in \{r_k, x, w_a, w_a'\}$. We now show that $w_i = x$ and, further, that since $w_i = x$, $T_k^k(i)$ is acyclic. We consider each claim in turn.

(i.) $w_i = x$

First observe that $w_i \neq r_k$ since $r_k w_i$ is an edge in T_i^k. Also, $w_i \neq w_a$ (this property is established by (R3) and was discussed in Section 2.1.5). Lastly, recall from (2.9) that $\varphi(v_i r_i) = \varphi(r_k w_i)$. So if $w_i = w_a'$ then $\varphi(v_i r_i) = \varphi(r_k w_a')$, contradicting (R6). Therefore, $w_i \neq w_a'$ and it follows that $w_i = x$.

(ii.) $T_k^k(i)$ is acyclic

Observe that since $w_i = x$, $w_i \in V(K_{2m})\backslash(\{\bigcup_{a=1}^{i-1} w_a, w'_a\} \cup \{r_k\})$ and w_i is a leaf adjacent to r_k in $T_k^k(i-1)$. Now, in order for $w_i w'_i$ to create a cycle in $T_k^k(i)$, there would have to exist a path from w_i to w'_i in $T_k^k(i-1)^*$. But, as we just observed, w_i is a leaf in $T_k^k(i-1)$ and since $T_k^k(i-1)^* = T_k^k(i-1) - r_k w_i$, w_i is an isolated vertex in $T_k^k(i-1)^*$ so it follows that no such path exists. Therefore, $T_k^k(i)$ is acyclic, as required.

The above two arguments show that (P12) holds for $T_k^k(i)$.

2.3.2 Property (P13)

In Section 2.3.1 above, we showed that $T_k^k(i)$ is acyclic for $1 \le i < k$. Recall from (2.11) that $T_k^k = T_k^k(k-1) - r_k w_{k-1} + w_k w'_l$. Thus, in order to prove T_k^k is acyclic, we need only show that adding $w_k w'_k$ to $T_k^k(k-1) - r_k w_k$ does not create a cycle. As in Section 2.3.1, let $T_k^k(k-1)^* = T_k^k(k-1) - r_k w_k$.

Observe from (2.11) that all of the edges of $T_k^k(k-1)$ are of the form $r_k x$, $r_k w'_i$ and $w_a w'_a$, where $1 \le i < k$ and $x \in V(K_{2m})\backslash(\{\bigcup_{a=1}^{k-1} w_i, w'_i\} \cup \{r_k\})$. Thus, $w_k \in \{r_k, x, w_i, w'_i\}$. We claim that $w_k = x$ and, further, that since $w_k = x$, T_k^k is acyclic. We consider each claim in turn.

(i.) $w_k = x$

Begin by observing that $w_k \ne r_k$ (since by (2.8) w_k and r_k were chosen to be distinct vertices) and, for $1 \le i < k$, $w_k \ne w_i$ (this property was established by (R5) and discussed in Section 2.2.1). The following argument shows $w_k \ne w'_i$.

First, observe that $w_k \ne w'_1$ since $\varphi(w_1 w'_1) = \varphi(r_k w_k)$, so if $w_k = w'_1$ then $w_1 = r_k$, which we know from (2.9) cannot be the case.

Now, for $2 \le i < k$, let $\alpha \in V(K_{2m})$ be the vertex such that $\varphi(w_k \alpha) = \varphi(r_k w_{i-1})$ and recall from (2.12) that $\varphi(w_i w'_i) = \varphi(r_k w_{i-1})$. Suppose that $w_k = w'_i$. Then since $\varphi(w_k \alpha) = \varphi(r_k w_{i-1}) = \varphi(w_i w'_i) = \varphi(w_i w_k)$, α must equal w_i. But from (2.9), we have that $\varphi(v_ir_i) = \varphi(r_kw_i)$, so if $w_i = \alpha$ then $\varphi(v_ir_i) = \varphi(r_k\alpha)$, contradicting (R7) which ensures that $\varphi(v_ir_i) \neq \varphi(r_k\alpha)$, where α is the vertex such that $\varphi(w_k\alpha) = \varphi(r_kw_{i-1})$. Therefore, $w_k \neq w'_i$, $2 \leq i < k$.

Combining the above arguments, it is clear that $w_k = x$.

(ii.) T_k^k is acyclic

Observe that since $w_k = x$ where $x \in V(K_{2m})\backslash(\{\bigcup_{a=1}^{k-1} w_i, w'_i\} \cup \{r_k\})$, w_k is a leaf adjacent to r_k in $T_k^k(k-1)$. In order for $w_kw'_k$ to form a cycle in T_k^k, there would have to exist a path from w_k to w'_k in $T_k^k(k-1)^*$. But because w_k is a leaf adjacent to r_k in $T_k^k(k-1)$, w_k is an isolated vertex in $T_k^k(k-1)^*$ since $T_k^k(k-1)^* = T_k^k(k-1) - r_kw_k$. It follows that no such path from w_k to w'_k exists in $T_k^k(k-1)^*$ and, consequently, T_k^k must be acyclic, as required.

It follows that T_k^k is acyclic, satisfying (P13).

The above Sections 2.3.1 and 2.3.2 show that properties (P12) and (P13) hold, thus completing the proof of the theorem. □

It is worth mentioning here that Theorem 1.1 guarantees the existence of $\left\lfloor \frac{\sqrt{6m+9}}{3} \right\rfloor$ mutually edge-disjoint rainbow spanning trees, but our algorithm can at times provide more such trees. The interested reader can see such an example in Appendix A.

Chapter 3

Problem 2: Balanced Sampling Designs Excluding Contiguous Units: A Complete Solution for $\lambda = 3$

Recall from Chapter 1 that a 2-BSEC(m, n, k, λ), $m, n \geq 3$, is a pair (X, B) where $X = \mathbb{Z}_m \times \mathbb{Z}_n$ and B is a collection of k-subsets of X (called blocks) such that each pair of 2-contiguous points do not appear together in any block, while any other two points appear together in exactly λ blocks.

As stated in Chapter 1, our result extends Theorem 1.4, solving the case where $\lambda = 3$. For convenience, we restate it here.

Theorem 1.5. *A 2-BSEC$(m, n, 3, 3)$ exists if and only if m and n are odd.*

Before proving our main result, we first prove a series of lemmas, demonstrating the existence of certain 2-BSEC$(m, n, 3, \lambda)$'s, before utilizing those lemmas in the proof of Theorem 1.5.

3.1 Constructing 2-BSEC$(m, n, 3, 3)$'s

First, we consider the case when $m \equiv n \equiv 1 \pmod 6$.

We will use the following well known combinatorial designs (see [16], for example for the results in this paragraph). A triple system, TS(n, λ), of order n and index λ is an ordered pair (S, T) where S is a finite set of n symbols and T is a collection of 3-element subsets of S called triples, such that each pair of distinct elements in S occurs together in exactly λ triples in T.

$$\begin{aligned}&\text{It is well known that there exists a TS}(n,\lambda) \text{ if } n \equiv 1 \text{ or } 3 \pmod 6 \text{ and } \lambda = 1,\\ &\text{and if } n \text{ is odd and } \lambda = 3.\end{aligned} \tag{3.1}$$

A quasigroup of order n is a pair $(Q, \circ)$, where Q is a set of size n and $\circ$ is a binary operation on Q such that for every pair of elements $a, b \in Q$, the equations $a \circ x = b$ and $y \circ a = b$ have unique solutions. A quasigroup is said to be idempotent if $i \circ i = i$ for $1 \le i \le n$.

Idempotent quasigroups are well known to exist for all orders of $n \neq 2$. (3.2)

A symmetric idempotent quasigroup is an idempotent quasigroup with the added restriction that for every $x, y \in Q$, $x \circ y = y \circ x$.

Symmetric idempotent quasigroups are well known to exist for all odd n. (3.3)

We can use idempotent quasigroups and one-dimensional BSECs to construct a 2-BSEC$(m, n, 3, 3)$ when $m \equiv n \equiv 1 \pmod 6$ and $n, m \ge 13$ as Lemma 3.1 shows.

We note here that in all the constructions in this section, the vertex set is $\mathbb{Z}_m \times \mathbb{Z}_n$, which can be visualized as a two-dimensional array consisting of m columns and n levels. So the point (i, j) occurs in column i on level j. And, since the points are elements of $\mathbb{Z}_m \times \mathbb{Z}_n$, any arithmetic operation on the first and second coordinates of any point are reduced modulo m and n respectively. The pairs of points, (i_1, j_1) and (i_2, j_2) can naturally be described as horizontal ($j_1 = j_2$), vertical ($i_1 = i_2$), or diagonal pairs ($i_1 \neq i_2$ and $j_1 \neq j_2$).

Lemma 3.1. *If $m \equiv 1 \pmod 6$ and $n \equiv 1 \pmod 6$ with $m, n \ge 13$ then there exists a 2-BSEC$(m, n, 3, 3)$.*

Proof. The blocks are defined as follows.

1. For each $j \in \mathbb{Z}_n$, let $(\mathbb{Z}_m \times \{j\}, B_i) \in B$ be a 1-BSEC$(m, 3, 3)$. This exists by Theorem 1.2. These blocks include each of the noncontiguous horizontal pairs of points three times.

2. For each $i \in \mathbb{Z}_m$, let $(\{i\} \times \mathbb{Z}_n, C_j) \in B$ be a 1-BSEC$(n, 3, 3)$. This exists by Theorem 1.2. These blocks include each of the noncontiguous vertical pairs of points three times.

3. Let $(\mathbb{Z}_m, T)$ be a TS$(m, 1)$ and let $(\mathbb{Z}_n, \circ)$ be an idempotent quasigroup. These exist by (3.1) and (3.2). For each $\{a, b, c\} \in T$, with $a < b < c$, and for each $r, s \in \mathbb{Z}_n$ with $r \neq s$, let B contain three copies of the block $\{(a, r), (b, s), (c, r \circ s)\}$. These blocks include each of the diagonal pairs of points three times.

It follows that $(\mathbb{Z}_m \times \mathbb{Z}_n, B)$ is the required 2-BSEC$(m, n, 3, 3)$.

$\square$

In view of Lemma 3.1, when $m \equiv n \equiv 1 \pmod 6$ it remains to consider the case where $m \equiv 1 \pmod 6$ and $n = 7$. The construction in Lemma 3.2 requires much more care than the previous one. Instead of using only a TS$(m, 1)$ and an idempotent quasigroup to create triples containing all of the diagonal pairs of points, in this construction (as well as the construction in Lemma 3.7) we adopt a different approach. Diagonal distances between points are tracked individually to check that each pair occurs together in three blocks. Formally, for each $x \in \mathbb{Z}_m$ and $y \in \mathbb{Z}_n$, the pair $\{(x, y), (x + d, y + i)\}$ is said to have diagonal distance (d, i) if $d \in \{1, 2, ..., \frac{m-1}{2}\}$ and $i \in \{1, 2, ...n - 1\}$. Similarly, the pair $\{(x, y), (x + d, y)\}$ if $d \leq \frac{m-1}{2}$ and the pair $\{(x, y), (x, y + d)\}$ if $d \leq \frac{n-1}{2}$ are said to have horizontal and vertical distance d respectively.

Lemma 3.2. *If $m \equiv 1 \pmod 6$, $m \geq 7$ and $n = 7$, then there exists a 2-BSEC$(m, 7, 3, 3)$.*

Proof. Let the set of points be $\mathbb{Z}_m \times \mathbb{Z}_7$, where $m \geq 7$. The set of blocks B is defined by taking the union of the following four sets of blocks, B_1, B_2, B_3, and B_4.

1. For each $x \in \mathbb{Z}_m$ and each $y \in \mathbb{Z}_7$, let $\{(x, y), (x, y + 2), (x, y + 4)\} \in B_1$. Blocks in B_1 include all vertical pairs of points distance 2 apart twice and distance 3 apart once.

2. For each $x \in \mathbb{Z}_m$ and each $y \in \mathbb{Z}_7$, let B_2 contain the following three triples:

(a) $\{(x, y), (x, y+2)(x + \frac{m-1}{2}, y+4)\}$,

(b) $\{(x, y), (x, y+3)(x + \frac{m-1}{2}, y+1)\}$, and

(c) $\{(x, y), (x, y+3)(x + \frac{m-1}{2}, y+6)\}$.

Blocks in B_2 include the remaining vertical pairs of points (distance 2 apart once and distance 3 apart twice), as well as all the pairs of points with diagonal distance $(\frac{m-1}{2}, i)$ for all level differences $i \in \{1, 2, ..., 6\}$ once (level differences 2 and 4 in (a), 1 and 5 in (b), and 3 and 6 in (c)).

3. For each $x \in \mathbb{Z}_m$, $y \in \mathbb{Z}_7$, for $2 \le j \le \frac{m-1}{2}$, and for $1 \le k \le 3$,

   (a) If j is odd then let $\{(x, y), (x+j, y), (x + \frac{m+j}{2}, y+k)\} \in B_3$, and

   (b) If j is even then let $\{(x, y), (x+j, y), (x + \frac{j}{2}, y+k)\} \in B_3$.

   Blocks in B_3 include all horizontal distances three times and all the pairs of points with diagonal distance (d, i) with $d < \frac{m-1}{2}$ and level differences $i \in \{1, 2, ..., 6\}$ once.

4. Let $(\mathbb{Z}_m, T)$ be a TS$(m, 1)$ and let $(\mathbb{Z}_7, \circ)$ be an idempotent quasigroup. These exist by (3.1) and (3.2). For each $\{a, b, c\} \in T$ with $a < b < c$, and for each $r, s \in \mathbb{Z}_7$ with $r \neq s$, let B_4 contain two copies of $\{(a, r), (b, s), (c, r \circ s)\}$.

   Blocks in B_4 include all of the diagonal pairs of points twice.

It follows that $(\mathbb{Z}_m \times \mathbb{Z}_n, B)$ is the required 2-BSEC$(m, 7, 3, 3)$.

□

This completes the proof of the case when $n \equiv m \equiv 1 \pmod 6$, so we now turn to the case when $m \equiv n \equiv 5 \pmod 6$. If m and n are at least 11 then we can construct a 2-BSEC$(m, n, 3, 3)$ as follows.

Lemma 3.3. *If $m \equiv n \equiv 5 \pmod 6$ and $m, n \geq 11$, then there exists a 2-BSEC$(m, n, 3, 3)$.*

Proof. Form the required 2-BSEC$(m, n, 3, 3)$, $(\mathbb{Z}_m \times \mathbb{Z}_n, B)$, as follows.

1. For each $j \in \mathbb{Z}_n$, let $(\mathbb{Z}_m \times \{j\}, B_i) \in B$ be a 1-BSEC$(m, 3, 3)$. This exists by Theorem 1.2. These blocks include each of the noncontiguous horizontal pairs of points three times.

2. For each $i \in \mathbb{Z}_m$, let $(\{i\} \times \mathbb{Z}_n, C_j) \in B$ be a 1-BSEC$(n, 3, 3)$. This exists by Theorem 1.2. These blocks include each of the noncontiguous vertical pairs of points three times.

3. Let $(\mathbb{Z}_m, T)$ be a TS$(m, 3)$ and let $(\mathbb{Z}_n, \circ)$ be a symmetric idempotent quasigroup. These exist by (3.1) and (3.3). For each $\{a, b, c\} \in T$, with $a < b < c$, and for each $r, s \in \mathbb{Z}_n$ with $r \neq s$, let B contain the triple $\{(a, r), (b, s), (c, r \circ s)\}$. These blocks include each of the diagonal pairs of points three times.

It follows that $(\mathbb{Z}_m \times \mathbb{Z}_n, B)$ is the required 2-BSEC$(m, n, 3, 3)$.

□

In view of Lemma 3.3, it remains to consider the case where $m \equiv 5 \pmod 6$ and $n = 5$. First, we will consider the case where $n = 5$, $m \equiv 5 \pmod 6$, and $m \geq 23$. To do so, we will use Langford sequences, hooked Langford sequences, and extended Skolem sequences [2, 18].

A [hooked] Langford sequence of defect δ and length μ, L(μ, δ) [HL(μ, δ)] is a sequence $(l_1, l_2, ..., l_{2\mu})$ $[(l_1, l_2, ..., l_{2\mu+1})]$ of 2μ $[2\mu + 1]$ integers with the property that: for every $k \in \{\delta, \delta + 1, ..., \delta + \mu - 1\}$, there exists a unique $i \in \{1, ..., 2\mu\}$ such that $l_i = l_{i+k} = k$ [and $l_{2\mu} = 0$].

For example, $(4, 2, 3, 2, 4, 3)$ is an L$(3, 2)$ and $(6, 4, 2, 5, 2, 4, 6, 3, 5, 0, 3)$ is an HL$(5, 2)$.

Theorem 3.4. [18] *A Langford sequence of defect δ and length μ exists if and only if*

1. $\mu \geq 2\delta - 1$ *and*

2. $\mu \equiv 0$ *or* $1 \pmod 4$ *for δ odd and* $\mu \equiv 0$ *or* $3 \pmod 4$ *for δ even.*

Theorem 3.5. [18] *A hooked Langford sequence of defect δ and length μ exists if and only if*

1. $\mu(\mu + 1 - 2\delta) + 2 \geq 0$ *and*

2. $\mu \equiv 2$ *or* 3 *(mod* 4*) for* δ *odd and* $\mu \equiv 1$ *or* 2 *(mod* 4*) for* δ *even.*

Note that this implies that for any δ, given large enough μ, we can find either a Langford or a hooked Langford sequence.

An extended Skolem sequence of length h, ES(h, p), is a sequence $(s_1, s_2, ..., s_{2h+1})$ of $2h+1$ integers such that for every $j \in \{1, 2, ..., h\}$, there exists a unique $i \in \{1, 2, ..., 2h+1\}$ such that $s_i = s_{i+j} = j$ and a unique p such that $s_p = 0$.

For example, $(3, 1, 1, 3, 4, 5, 0, 2, 4, 2, 5)$ is an ES$(5, 7)$.

Note that an extended Skolem sequence with $s_{2h} = 0$ is a hooked Langford sequence of defect 1 and length h.

Theorem 3.6. [2] *An extended Skolem sequence ES*(h, p) *exists if and only if* p *is odd and* $h \equiv 0$ *or* 1 *(mod* 4*) or* p *is even and* $h \equiv 2$ *or* 3 *(mod* 4*).*

Note that this implies that an extended Skolem sequence of order h exists for all h.

Lemma 3.7. *If* $m \equiv 5$ *(mod* 6*),* $m \geq 23$ *and* $n = 5$*, then there exists a* 2*-BSEC(*$m, 5, 3, 3$*).*

Proof. The required 2-BSEC$(m, 5, 3, 3)$, $(\mathbb{Z}_m \times \mathbb{Z}_5, B)$, is defined as follows.

1. For $0 \leq x \leq m-1$, $0 \leq y \leq 4$, $2 \leq i \leq \frac{m-1}{2}$, and $1 \leq j \leq 2$, let

   (a) $\{(x, y), (x+i, y), (x+\frac{i}{2}, y+j)\} \in B_1$ if i is even,

   (b) $\{(x, y), (x+i, y), (x+\frac{i+m}{2}, y+j)\} \in B_1$ if i is odd,

   (c) $\{(x, y), (x, y+2), (x+\frac{m-1}{2}, y+3)\} \in B_1$, and

   (d) $\{(x, y), (x, y+2), (x+\frac{m-1}{2}, y+4)\} \in B_1$.

   Blocks defined in (a) and (b) include each of the noncontiguous horizontal pairs of points twice and each of the pairs of points with diagonal distance (d, i) with $d < \frac{m-1}{2}$ and level difference $i \in \{1, 2, 3, 4\}$ once. Blocks defined in (c) and (d) include each of the noncontiguous vertical pairs of points twice and each of the pairs of points with diagonal distance $(\frac{m-1}{2}, i)$ with level difference $i \in \{1, 2, 3, 4\}$ once.

2. Let $\mu = \frac{\frac{m-1}{2}-2}{3}$.

   (a) First, suppose $\mu \equiv 0$ or $3 \pmod 4$ and let $L = (l_1, l_2, ..., l_{2\mu})$ be a Langford sequence $L(\mu, \delta = 2)$, which exists by Theorem 3.4. For each $k \in \{2, 3, ..., \mu + 1\}$, if $l_i = l_j = k$ with $i < j$, then let $l_{k,1} = i + \mu + \delta - 1 = i + \mu + 1$ and let $l_{k,2} = j + \mu + \delta - 1 = j + \mu + 1$. Notice that i and j represent the positions of k in the Langford sequence.

   Then, for $0 \leq x \leq m - 1$, $0 \leq y \leq 4$, and $2 \leq k \leq \mu + 1$, let $b_k = \{(x, y), (x + l_{k,1}, y), (x + l_{k,2}, y)\} \in B_2$. So, each b_k in B_2 contains three pairs of points, their horizontal distances being $l_{k,1}, l_{k,2}$, and $|l_{k,1} - l_{k,2}| = k$.

   (b) Now, suppose $\mu \equiv 1$ or $2 \pmod 4$ and let $HL = (l_1, l_2, ..., l_{2\mu+1})$ be a hooked Langford sequence $HL(\mu, \delta = 2)$, which exists by Theorem 3.5. For each $w \in \{2, 3, ..., \mu + 1\}$, if $l_u = l_v = w$ with $u < v$, let $l_{w,1} = u + \mu + \delta - 1 = u + \mu + 1$ and $l_{w,2} = v + \mu + \delta - 1 = v + \mu + 1$.

   Notice that u and v represent the positions of w in the hooked Langford sequence.

   Then, for $0 \leq x \leq m - 1$, $0 \leq y \leq 4$, and $2 \leq w \leq \mu + 1$, let $b_w = \{(x, y), (x + l_{w,1}, y), (x + l_{w,2}, y)\} \in B_2$. So each b_w in B_2 contains three pairs of points, their horizontal distances being $l_{w,1}, l_{w,2}$, and $|l_{w,1} - l_{w,2}| = w$.

   Blocks defined in (a) include each of the noncontiguous horizontal pairs of points once, except distance $d = \frac{m-1}{2}$. Blocks defined in (b) include each of the noncontiguous horizontal pairs of points once, except distance $d = \frac{m-1}{2} - 1$.

3. Again, let $\mu = \frac{\frac{m-1}{2}-2}{3}$ and we will consider two cases depending on whether $\mu \equiv 0$ or $3 \pmod 4$ or $\mu \equiv 1$ or $2 \pmod 4$.

(a) If $\mu \equiv 0$ or 3 (mod 4), then for $0 \leq x \leq m-1$ and $0 \leq y \leq 4$, let B_3 contain the blocks:

i. if $3\mu + 2$ is even:

A. $\{(x, y), (x + 3\mu + 2, y), (x + \frac{3\mu+2}{2}, y + 2)\}$ and

B. $\{(x, y), (x, y + 2), (x + \frac{3\mu+2}{2}, y + 1)\}$; or

ii. if $3\mu + 2$ is odd:

A. $\{(x, y), (x + 3\mu + 2, y), (x + \frac{3\mu+2+m}{2} y + 2)\}$ and

B. $\{(x, y), (x, y + 2), (x + \frac{m-(3\mu+2)}{2}, y + 1)\}$

(b) If $\mu \equiv 1$ or 2 (mod 4), then for $0 \leq x \leq m-1$ and $0 \leq y \leq 4$, let B_3 contain the blocks:

i. if $3\mu + 1$ is even:

A. $\{(x, y), (x + 3\mu + 1, y), (x + \frac{3\mu+1}{2}, y + 2)\}$ and

B. $\{(x, y), (x, y + 2), (x + \frac{3\mu+1}{2}, y + 1)\}$; or

ii. if $3\mu + 1$ is odd:

A. $\{(x, y), (x + 3\mu + 2, y), (x + \frac{3\mu+2+m}{2}, y + 2)\}$ and

B. $\{(x, y), (x, y + 2), (x + \frac{m-3\mu+1}{2}, y + 1)\}$.

Blocks defined in (a) include each of the horizontal pairs of points with distance $\frac{m-1}{2}$ once, each of the noncontiguous vertical pairs of points once, and each of the pairs of points with diagonal distance $(\frac{m-1}{4}, i)$ with level distance $i \in \{1, 2, 3, 4\}$ twice. Blocks defined in (b) include each of the horizontal pairs of points with distance $\frac{m-1}{2} - 1$ once, each of the noncontiguous vertical pairs of points once, and each of the pairs of points with diagonal distance $(\frac{m+1}{4} - 1, i)$ with level distance $i \in \{1, 2, 3, 4\}$ twice.

4. Lastly, let $h = \lfloor \frac{m-1}{3} \rfloor$ and $p = \frac{m-1}{2} - 1$. Let $ES = (s_1, s_2, ..., s_{2h+1})$ be an extended Skolem sequence $ES(h, p)$. Notice that h is odd since $m \equiv 5$ (mod 6). Furthermore,

if $h \equiv 1 \pmod 4$ then p is odd and if $h \equiv 3 \pmod 4$ then p is even. Therefore by Theorem 3.6 ES exists.

If $s_i = s_{i+j} = j$ with $i < i+j$, let $s_{j,1} = i+h$ and $s_{j,2} = i+j+h$. Notice that i and $i+j$ represent the positions of $j \in \{1, 2, ..., h\}$.

Now, let $(\mathbb{Z}_5, \circ)$ be an idempotent quasigroup, which exists by (3.2). For each $q, r \in \mathbb{Z}_5$ with $q \neq r$, let two copies of $\{(x, q), (x + s_{j,1}, r), (x + s_{j,2}, q \circ r)\} \in B_4$.

These blocks include each of the pairs of points with diagonal distance (d, i) with $d < \frac{m+1}{4}$ or $\frac{m-1}{4} - 1$, depending on whether $\mu \equiv 0, 3 \pmod 4$ or $\mu \equiv 1, 2 \pmod 4$ respectively, and with level difference $i \in \{1, 2, 3, 4\}$.

Let $B = B_1 \cup B_2 \cup B_3 \cup B_4$. It follows that $(\mathbb{Z}_5 \times \mathbb{Z}_m, B)$ is the required 2-BSEC$(m, 5, 3, 3)$.

$\square$

Lastly, we consider the following three cases where $n = 5$ and $m = 5, 11$, and 17.

Lemma 3.8. *If $n = 5$ and $m \in \{5, 11, 17\}$ then there exists a 2-BSEC$(m, 5, 3, 3)$.*

Proof. We consider each of the three values of m in turn.

1. $m = n = 5$.

   For $x \in \mathbb{Z}_5$ and $y \in \mathbb{Z}_5$, let B contain one copy of the following:

   $\{(x, y), (x + 2, y), (x, y + 2)\}, \{(x, y), (x + 2, y), (x + 1, y + 1)\}$,

   $\{(x, y), (x + 2, y), (x + 2, y + 2)\}, \{(x, y), (x + 1, y + 1), (x, y + 2)\}$,

   $\{(x, y), (x + 1, y + 1), (x + 3, y + 2)\}, \{(x, y), (x + 3, y + 1), (x + 2, y + 2)\}$,

   $\{(x, y), (x + 1, y + 2), (x + 2, y + 4)\}, \{(x, y), (x + 4, y + 2), (x + 3, y + 4)\}$,

   $\{(x, y), (x + 1, y + 2), (x + 3, y + 4)\}, \{(x, y), (x + 3, y + 2), (x + 2, y + 4)\}$.

   Then $(\mathbb{Z}_5 \times \mathbb{Z}_5, B)$ is the required 2-BSEC$(5, 5, 3, 3)$.

2. $m = 11$, $n = 5$.

   For each $j \in \mathbb{Z}_5$ let $(\mathbb{Z}_{11} \times \{j\}, B_j)$ be a 1-BSEC$(11, 3, 3)$.

   For each $x \in \mathbb{Z}_{11}$ and $y \in \mathbb{Z}_5$ let B_5 contain one copy of the following:

   $\{(x, y), (x + 2, y + 1), (x + 5, y + 2)\}, \{(x, y), (x + 2, y + 3), (x + 9, y + 1)\},$

   $\{(x, y), (x + 1, y + 2), (x + 7, y + 4)\}, \{(x, y), (x + 2, y + 2), (x + 5, y + 4)\},$

   $\{(x, y), (x + 2, y + 3), (x + 8, y + 2)\}, \{(x, y), (x + 4, y + 1), (x + 5, y + 3)\},$

   $\{(x, y), (x + 2, y + 1), (x + 8, y + 2)\}, \{(x, y), (x + 7, y + 1), (x + 4, y + 2)\},$

   $\{(x, y), (x + 9, y + 1), (x + 3, y + 2)\}, \{(x, y), (x + 5, y + 2), (x + 1, y + 4)\},$

   $\{(x, y), (x + 2, y + 2), (x + 8, y + 4)\}, \{(x, y), (x + 4, y + 4), (x + 7, y + 2)\},$

   $\{(x, y), (x + 4, y + 2), (x + 7, y + 1)\}, \{(x, y), (x + 3, y + 2), (x + 5, y + 1)\},$

   $\{(x, y), (x + 3, y + 1), (x + 9, y + 2)\}, \{(x, y), (x + 2, y + 2), (x + 7, y + 4)\},$

   $\{(x, y), (x + 1, y + 2), (x + 2, y + 1)\}, \{(x, y), (x + 7, y + 2), (x + 10, y + 1)\},$

   and three copies of:

   $\{(x, y), (x, y + 2), (x + 1, y + 3)\}.$

   Then $(\mathbb{Z}_{11} \times \mathbb{Z}_5, \cup_{i \in \mathbb{Z}_6} B_i)$ is the required 2-BSEC$(11, 5, 3, 3)$.

3. $n = 5$, $m = 17$

   This design was found using a hill climbing algorithm. Rather than listing all of the triples here, the interested reader can see such a design in Appendix B.

This completes the proof of Lemma 3.8. □

3.2 The Main Result

We are now ready to provide the following necessary and sufficient conditions for a 2-BSEC$(m, n, 3, 3)$ to exist.

Theorem 3.9. *A 2-BSEC($m, n, 3, 3$) exists if and only if m and n are odd.*

Proof. The necessity follows from Lemma 1.3. To prove sufficiency, the necessary conditions together with the symmetry of m and n, mean that the only cases that need to be considered are:

1. $m \equiv 1 \pmod 6$ and $n \equiv 1, 3, 5 \pmod 6$,
2. $m \equiv 3 \pmod 6$ and $n \equiv 3 \pmod 6$, and
3. $m \equiv 5 \pmod 6$ and $n \equiv 1, 3, 5 \pmod 6$.

By Theorem 1.4, the existence of a 2-BSEC($n, m, 3, 1$) is established for the following cases:

1. $m \equiv 1 \pmod 6$ and $n \equiv 3, 5 \pmod 6$,
2. $m \equiv 3 \pmod 6$ and $n \equiv 3 \pmod 6$, and
3. $m \equiv 5 \pmod 6$ and $n \equiv 1, 3 \pmod 6$.

In each of the cases 1-3, the blocks in the 2-BSEC($m, n, 3, 1$) can be repeated three times to produce a 2-BSEC($m, n, 3, 3$). Therefore, the only cases remaining to be considered are:

1. $m \equiv 1 \pmod 6$ and $n \equiv 1 \pmod 6$ and
2. $m \equiv 5 \pmod 6$ and $n \equiv 5 \pmod 6$.

We can assume that $n \leq m$. The first case is settled in Lemma 3.2 if $n = 7$ and in Lemma 3.1 otherwise. The second case is settled in Lemmas 3.7 and 3.8 if $n = 5$ and in Lemma 3.3 otherwise, completing the proof of the theorem.

□

Chapter 4

Other Directions

Both of the problems discussed in this dissertation can be extended or expanded upon in other directions. The following sections contain a discussion on a few ways to do this.

4.1 Rainbow Trees

In the introduction, we looked at three conjectures related to finding edge-disjoint rainbow spanning trees in properly edge-colored complete graphs: Conjectures 1.1, 1.2, and 1.3, before focusing specifically on Conjecture 1.1 for the remainder of our discussion.

The first step in extending the research discussed in this dissertation would be to improve the result obtained in Theorem 1.1, with the goal of fully proving Conjecture 1.1. There is also significant room for improvement to the results obtained so far pertaining to Conjecture 1.3.

Another direction would be to look at the number of edge-disjoint rainbow spanning trees or rainbow spanning uni-cyclic graphs (rainbow graphs with exactly one cycle) in a properly edge-colored K_{2m-1}. Since K_{2m-1} has $(2m-1)(m-1)$ edges, we conjecture the following.

Conjecture 4.1. *If K_{2m-1} is $(2m-1)$-edge-colored, then the edges of K_{2m-1} can be partitioned into $m-1$ rainbow spanning trees together with one near-perfect matching containing the $m-1$ edges of a single color class.*

Conjecture 4.2. *If K_{2m-1} is $(2m-1)$-edge-colored, then the edges of K_{2m-1} can be partitioned into $m-1$ rainbow spanning trees together with one near-perfect rainbow matching containing $m-1$ edges.*

Conjecture 4.3. *If K_{2m-1} is $(2m-1)$-edge-colored, then the edges of K_{2m-1} can be partitioned into $m-1$ rainbow spanning uni-cyclic graphs.*

4.2 Balanced Sampling Plans Excluding Contiguous Units

As discussed in the introduction, significant work has been done considering balanced sampling plans excluding contiguous units in one or two dimensions, but little to no work has been done in three or more dimensions. To do so, we can generalize the definition of 2-contiguous points to n-contiguous in the following way.

Given a set of points $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times ... \times \mathbb{Z}_{m_n}$ arranged in n dimensions, the n-contiguous points to a point $(x_1, x_2, ..., x_n)$ are the points $(y_1, y_2, ..., y_n)$ where for coordinate i, $1 \leq i \leq n$, $y_i = x_i + 1$ or $y_i = x - 1$, reducing the sums mod m_i, and for $1 \leq j \leq n$, $j \neq i$, $x_j = y_j$.

We can also generalize balanced sampling plans excluding contiguous units to n dimensions. Define an n-BSEC$(m_1, m_2, ..., m_n, k, \lambda)$ to be a pair (X, B) where $X = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times ... \times \mathbb{Z}_{m_n}$ and B is a collection of k-subsets of X (called blocks) such that each pair of n-contiguous points do not appear together in any block, while any other two points appear together in exactly λ blocks.

Armed with these definitions, we can now ask the question: for which values of $m_1, m_2, ..., m_n, k$, and λ does an n-BSEC$(m_1, m_2, ..., m_n, k, \lambda)$ exist.

References

[1] S. Akbari, A. Alipour, H.-L. Fu and Y.-H. Lo, Multicolored parallelism of isomorphic spanning trees, *SIAM J. Discrete Math.*, 20(2006), No. 3, 564–567.

[2] C.A. Baker, Extended Skolem Sequences. Journal of Combinatorial Designs **1995**, *3:5*, 363-379.

[3] D. Banks, G. Constantine, A. Merriwether and R. LaFrance, Nonparametric inference on mtDNA mismatches, *J. Nonparametre. Statist.*, 11(1999), 215–232.

[4] R. A. Brualdi and S. Hollingsworth, Multicolored trees in complete graphs, *J. Combin. Theory Ser. B*, 68 (1996), No. 2, 310–313.

[5] Bryant, D., Chang, Y., Rodger, C.A., Wei, R. Two-Dimensional Balanced Sampling Plans Excluding Contiguous Units. Communications in Statistics - Theory and Methods **2002**, *31:8*, 1441-1455.

[6] J. Carraher, S. Hartke, and P. Horn, Edge-disjoint rainbow spanning trees in complete graphs, *European J. Combin.*, 57(2016), 71–84.

[7] Colbourn, C.J., Ling, A.C.H. A Class of Partial Triple Systems with Applications in Survey Sampling. Commun. Statis. Theory Methods **1998**, *27*, 1009-1018.

[8] G. M. Constantine, Multicolored parallelisms of isomorphic spanning trees, *Discrete Math. Theor. Comput. Sci.*, 5(2002), No. 1, 121-125.

[9] T. Feng, X. Wang, Y. Xu, J. Zhang, Two-dimensional balanced sampling plans avoiding adjacent units, *Discrete Mathematics*, 338(2015), No. 9, 1624-1642.

[10] H.-L. Fu and Y.-H. Lo, Multicolored isomorphic spanning trees in complete graphs, *Ars Combin.*, 122(2015), 423–430.

[11] F. Harary, Parallel concepts in graph theory, *Math. Comput. Modeling*, 18(1993), 101–105.

[12] Hedayat, A.S., Rao, C.R., Stufken, J. Designs in Survey Sampling Avoid Contiguous Units. *Handbook of Statistics*; Krishnaiah, P.R., Rao, C.R., Eds.; Elsevier Science Publishers B.V. **1988**, Vol. 6, 575-583.

[13] Hedayat, A.S., Rao, C.R., Stufken, J. Sampling Plans Excluding Contiguous Units. J. Statis. Plann. Infer. **1988**, *19*, 159-170.

[14] M. Jacroux, On the construction of sets of integers with equal power sums, *J. Number Theory*, 52(1995), No. 1, 35–42.

[15] J. Krussel, S. Marshall and H. Verrall, Spanning Trees Orthogonal to One-Factorizations of K_{2n}, *Ars Combin.*, 57(2000), 77–82.

[16] Lindner, C.C., Rodger, C.A. *Design Theory*; CRC Press, **1997**.

[17] See, K., Song, S.Y., Stufken J. On a Class of Partially Incomplete Block Designs with Applications in Survey Sampling. Comm. Statist. Theory Methods **1997**, *26*, 1-13.

[18] Simpson, J.E. Langford Sequences: Perfect and Hooked. Discrete Math **1983**, *44*, 97-104.

[19] Stufken, J. Combinatorial and Statistical Aspects of Sampling Plans to Avoid the Selection of Adjacent Units. J. Comb. Inf. Sys. Science **1993**, *18*, 149-160.

Appendices

Appendix A

Edge-Disjoint Rainbow Spanning Trees in K_{14}

This appendix demonstrates the use of our algorithm on a properly edge-colored K_{14}. In this instance, $m = 7$ and Theorem 1.1 guarantees the existence of $\Omega_7 = \left\lfloor \frac{\sqrt{6*7+9}}{3} \right\rfloor = 2$ edge-disjoint rainbow spanning trees. As mentioned in Chapter 2, our algorithm ensures the existence of at least Ω_m trees, but can at times generate more. The following example constructs 3 mutually edge-disjoint rainbow spanning trees from K_{14} using our algorithm.

We begin with a given edge-coloring K_{14}.

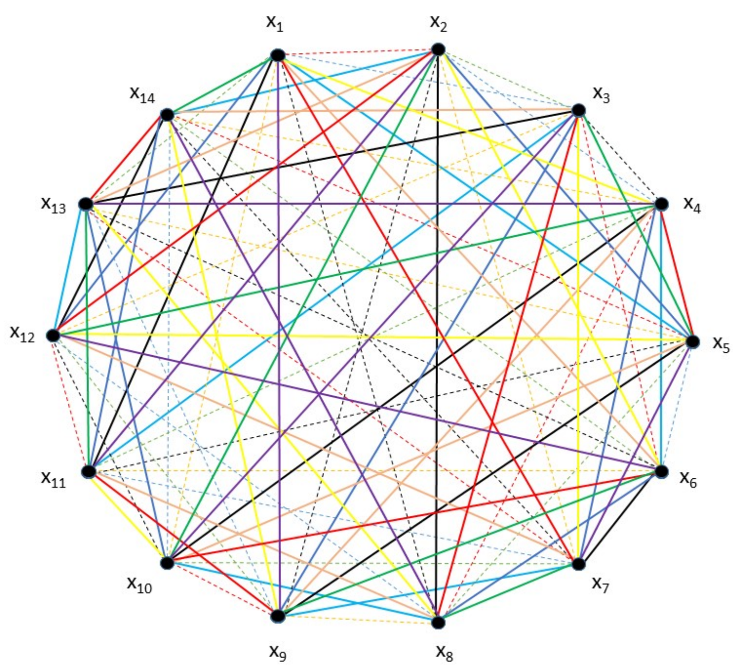


Figure A.1: A Proper Edge-Coloring of K_{14}

Step 1: $k = 1$

r_1 can be any vertex in $V(K_{14})$. Let $x_1 = r_1$ and then $T_1^1 = S_{x_1}$.

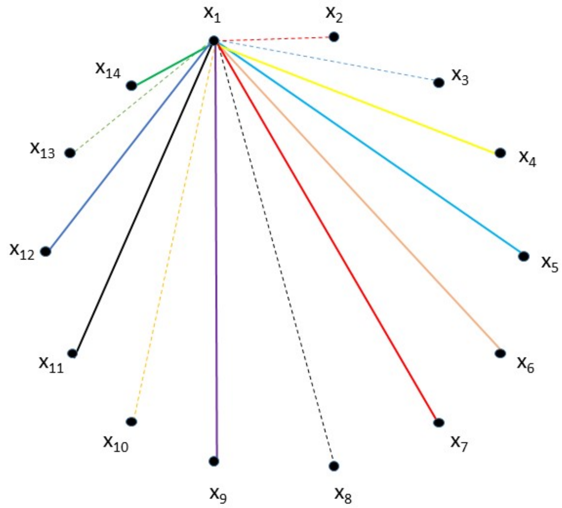


Figure A.2: T_1^1

Step 2: $k = 2$

Recall from (2.6) that we defined L_{k-1} to be the set of all vertices that are leaves adjacent to the root in the trees T_i^{k-1}, $1 \leq i < k$. Thus, when $k = 2$, $L_1 = V(K_{14}) \backslash \{x_1\}$, which has cardinality larger than $6 * 2 - 7 = 5$, as required. We now select distinct vertices r_2 and w_2^2 from L_1.

Let $r_2 = x_5$ and $w_2^2 = x_4$. Then $L_1^* = V(K_{14}) \backslash \{x_1, x_4, x_5\}$.

Our next step is to determine v_1^2. (R1) eliminates vertices x_1, x_4, and x_5 as choices for v_1^2. Items (R2 - R11) additionally eliminate vertices x_6, x_7, and x_9. Let $v_1^2 = x_2$. Then $v_1^{2'} = w_1^2 = x_14$ and we can form T_1^2 from T_1^1 by having $T_1^2 = T_1^1 - x_1x_5 - x_1x_2 + x_5x_{14} + x_2x_{14}$.

Forming T_1^2 allows us to form $T_2^2(1) = S_{x_5} - x_5x_{14} + x_{14}x_{13}$ where the color of edge $x_{14}x_{13}$ is red, like edge x_5x_4 and thus, $w_1^{2'} = x_{13}$.

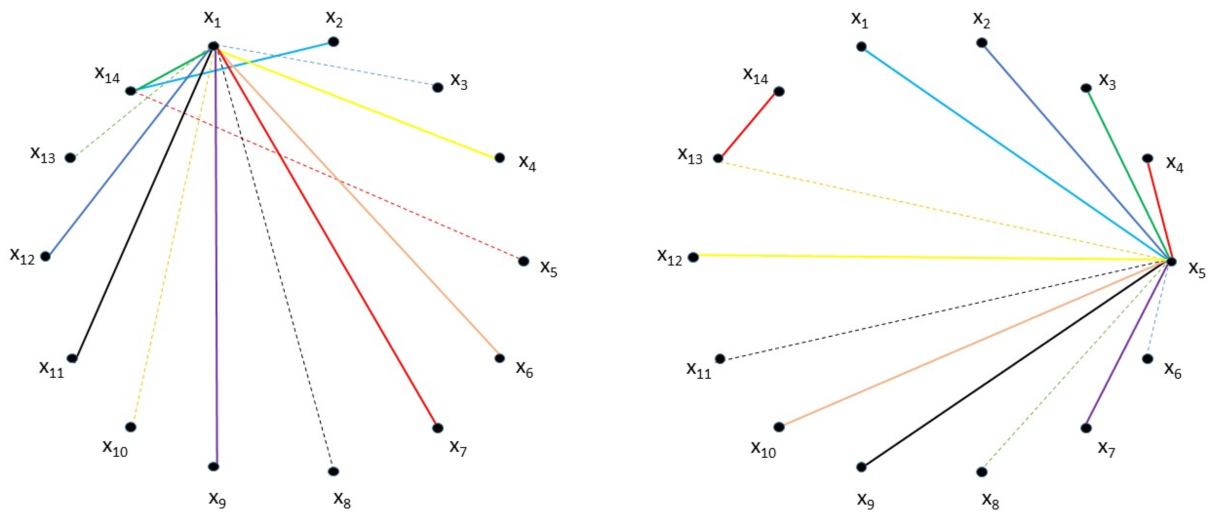


Figure A.3: T_1^2 on the left with $T_2^2(1)$ on the right

$T_2^2(2)$ can then be formed from $T_2^2(1)$ by letting $T_2^2(2) = S_{x_5} - x_5x_{14} - x_5x_4 + x_{14}x_{13} + x_4x_8$ where $w_2^{2'} = x_8$.

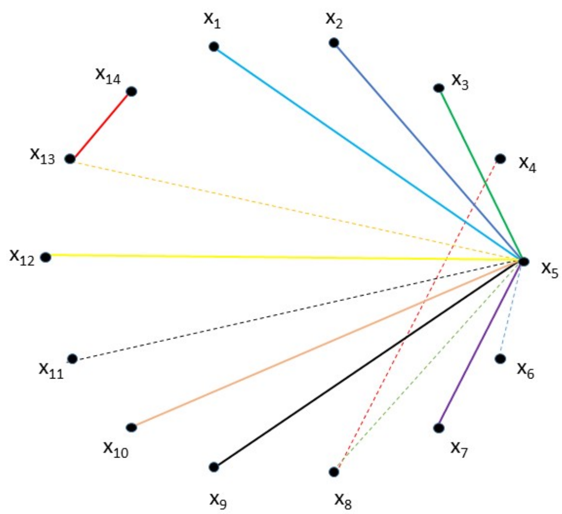


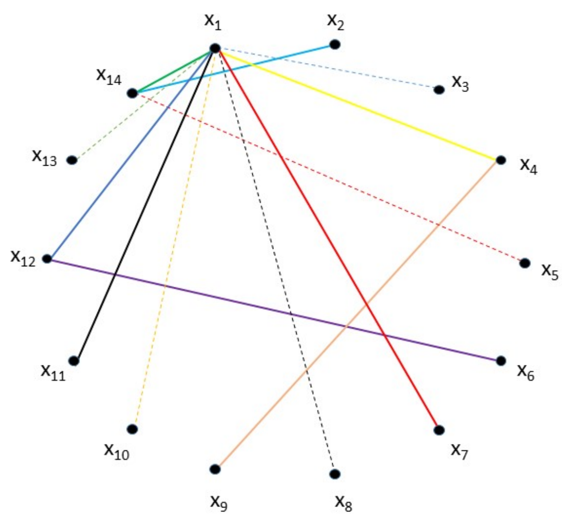
Figure A.4: T_2^2

Step 3: $k = 3$

At this point $L_2 = L_1 \backslash \{x_2, x_4, x_5, x_8, x_{13}, x_{14}\}$ so $|L_2| = 7$, which is not greater than the $6 * 3 - 7 = 11$ vertices necessary for our algorithm to guarantee a third tree. However, we can still find suitable vertices r_3 and w_3^3 that allow for three edge-disjoint rainbow spanning trees to be formed.

Let $r_3 = x_9$ and $w_3^3 = x_{11}$. Then $L_2^* = \{x_3, x_6, x_7, x_{10}, x_{12}\}$ and we can find suitable v_1^3 and v_2^3 in L_2^*, as described below.

In addition to the restriction that $v_1^3 \in L_2^*$, items (R2 - R11) additionally eliminate vertices x_3 and x_7 as candidates for v_1^3. We thus let $v_1^3 = x_6$ and then $w_1^3 = x_4$, $v_1^{3'} = x_{12}$, and $w_1^{3'} = x_3$, allowing us to form both $T_1^3 = T_1^2 - x_1x_9 - x_1x_6 + x_9x_4 + x_6x_{12}$ and $T_3^3(1) = S_{x_9} - x_9x_4 + x_4x_5$.



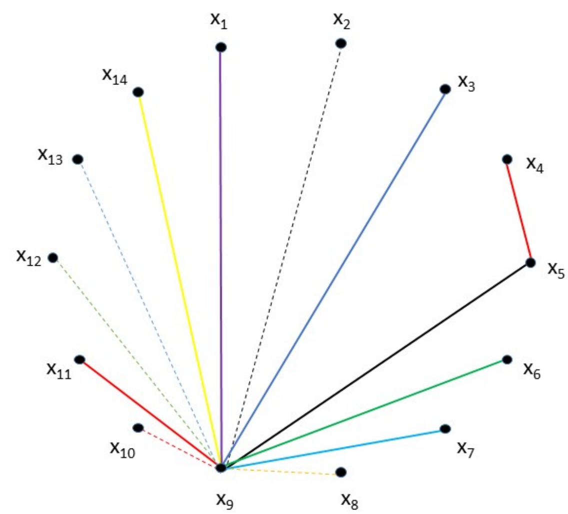


Figure A.5: T_1^3 on the left with $T_3^3(1)$ on the right

Similarly, $v_2^3 \in L_2^*$ and items (R2 - R11) additionally eliminate vertex x_{10} as a candidate for v_2^3. So we can let $v_2^3 = x_{12}$ and then $v_2^{3'} = w_2^3 = x_{14}$ and $w_2^{3'} = x_3$, allowing us to form both $T_2^3 = T_2^2 - x_5x_9 - x_5x_{12} + x_9x_{14} + x_{12}x_{14}$ and $T_3^3(2) = S_{x_9} - x_9x_4 - x_9x_{14} + x_4x_5 + x_{14}x_3$.

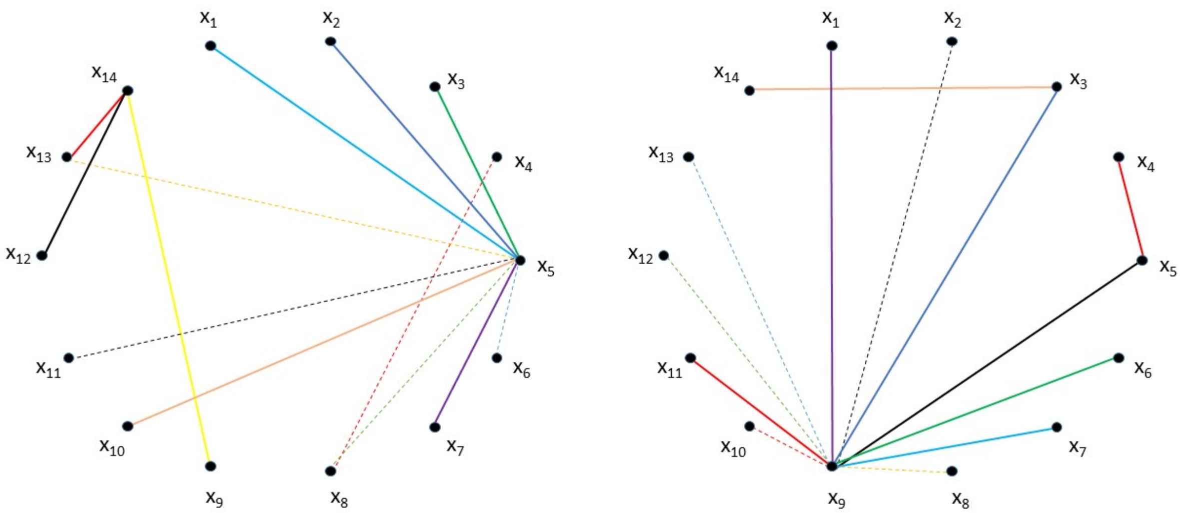


Figure A.6: T_2^3 on the left with $T_3^3(2)$ on the right

Lastly, forming $T_3^3(2)$ determines $w_3^{3'}$, thus allowing us to form the third tree, $T_3^3 = S_{x_9} - x_9x_4 - x_9x_{14} - x_9x_{11} + x_4x_5 + x_{14}x_3 + x_{11}x_{10}$.

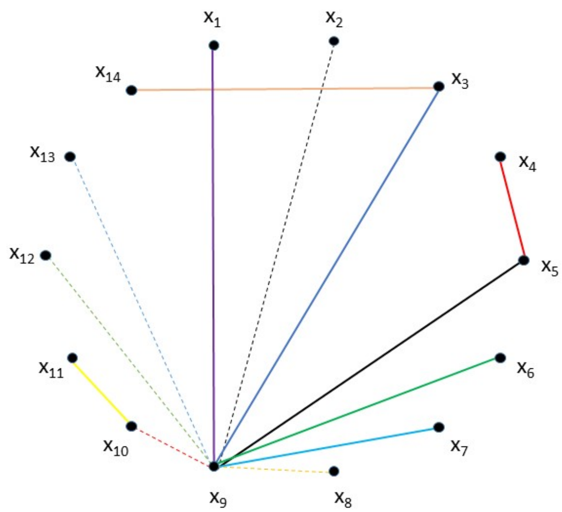


Figure A.7: T_3^3

At this point, $L_3 = \{x_7\}$ and we cannot find distinct vertices r_4 and w_4^4 in L_3 to possibly create a fourth tree using our algorithm.

Appendix B

The Remaining 2-BSEC$(m, n, 3, 3)$

For the remaining case of Lemma 3.8, a 2-BSEC$(17, 5, 3, 3)$, we used a hill climbing algorithm to find such a balanced sampling plan. The program was written in the Java programming language and implemented in the jGRASP environment. We first created a two-dimensional 85×85 adjacency matrix and then placed 3 edges between non-contiguous pairs of points. The adjacency matrix represents a two-dimensional array with 5 rows and 17 columns where the rows contain the numbers $0-16$, $17-33$, $34-50$, $51-67$, and $68-84$. This code was able to produce multiple 2-BSEC$(17, 5, 3, 3)$'s, one of which is included below in Section B.1. The code is included in Section B.2.

B.1 A 2-BSEC$(17, 5, 3, 3)$

Table B.1: A 2-BSEC$(17, 5, 3, 3)$

A 2-BSEC$(17, 5, 3, 3)$				
28, 30, 79	21, 40, 56	13, 19, 22	24, 68, 75	45, 60, 67
29, 52, 66	14, 37, 72	16, 42, 50	26, 68, 74	13, 15, 55
56, 58, 76	51, 78, 82	25, 77, 81	0, 15, 26	17, 45, 71
26, 52, 68	31, 44, 64	13, 63, 79	51, 56, 61	20, 23, 62
54, 72, 80	18, 47, 57	24, 36, 84	10, 58, 65	29, 34, 53
3, 30, 58	30, 56, 83	31, 71, 76	19, 47, 63	2, 14, 26
11, 21, 43	25, 46, 66	45, 64, 82	8, 11, 48	3, 41, 72
3, 62, 67	15, 18, 79	10, 44, 73	21, 48, 75	20, 67, 69

1, 26, 45	1, 26, 39	6, 61, 83	20, 59, 81	28, 44, 62
11, 15, 50	17, 49, 78	18, 25, 65	44, 70, 79	8, 52, 78
0, 34, 44	27, 35, 59	7, 35, 61	32, 37, 66	4, 60, 65
17, 20, 43	58, 69, 71	32, 47, 67	10, 67, 82	19, 65, 81
43, 58, 70	29, 58, 62	18, 28, 38	2, 7, 44	18, 71, 84
2, 24, 30	4, 22, 35	8, 18, 54	7, 17, 47	17, 64, 72
1, 12, 47	11, 30, 65	30, 39, 58	22, 65, 67	6, 28, 31
15, 44, 73	41, 62, 78	21, 26, 80	0, 31, 75	31, 33, 66
13, 27, 45	37, 49, 75	28, 50, 51	19, 21, 73	12, 41, 61
7, 13, 51	54, 60, 68	3, 29, 55	8, 37, 72	45, 49, 68
1, 29, 41	27, 51, 82	6, 32, 79	2, 11, 62	1, 39, 63
34, 36, 80	20, 26, 64	58, 65, 77	14, 22, 55	14, 63, 78
12, 43, 72	0, 5, 55	9, 20, 66	16, 56, 68	35, 42, 77
7, 14, 36	15, 21, 71	17, 74, 77	33, 59, 63	0, 43, 63
45, 54, 81	5, 19, 25	31, 58, 79	20, 54, 63	50, 60, 70
17, 19, 24	12, 54, 61	8, 21, 66	33, 35, 37	27, 66, 72
18, 23, 69	32, 42, 52	30, 32, 68	23, 30, 38	23, 35, 54
11, 66, 68	36, 71, 79	30, 53, 55	14, 53, 69	24, 54, 81
29, 61, 73	21, 29, 50	30, 33, 44	35, 64, 82	31, 57, 67
24, 34, 52	22, 24, 68	45, 53, 73	7, 13, 21	20, 44, 48
12, 30, 79	8, 35, 54	25, 29, 82	3, 72, 75	12, 18, 67
6, 14, 81	4, 41, 78	14, 50, 75	23, 26, 49	35, 58, 82
33, 60, 71	1, 17, 75	5, 36, 41	20, 34, 73	10, 20, 62
26, 40, 79	5, 23, 57	2, 33, 68	27, 47, 49	48, 62, 66
20, 31, 75	9, 50, 60	35, 38, 81	58, 67, 73	68, 78, 83
30, 38, 70	25, 35, 80	1, 45, 79	4, 10, 33	6, 19, 77

18, 52, 70	3, 19, 41	15, 27, 53	35, 67, 79	50, 59, 66
26, 51, 63	12, 15, 19	58, 72, 74	4, 71, 73	4, 6, 38
24, 36, 52	21, 24, 61	45, 61, 74	44, 58, 74	19, 21, 49
2, 9, 67	27, 32, 67	40, 49, 84	19, 46, 76	6, 25, 72
25, 50, 73	16, 32, 74	21, 42, 77	57, 61, 71	0, 45, 76
3, 45, 78	11, 21, 37	5, 50, 69	37, 40, 58	17, 65, 80
21, 29, 59	41, 49, 56	16, 21, 43	14, 35, 41	23, 34, 75
12, 59, 70	56, 59, 64	64, 78, 84	35, 44, 67	5, 12, 39
21, 24, 68	33, 75, 81	30, 37, 55	2, 28, 76	46, 60, 69
20, 57, 77	5, 44, 81	30, 35, 50	0, 30, 82	22, 46, 56
1, 32, 58	7, 42, 63	29, 41, 54	8, 46, 62	12, 35, 75
9, 19, 53	32, 62, 66	9, 52, 79	6, 25, 30	15, 19, 47
14, 46, 51	46, 53, 71	0, 39, 80	10, 41, 72	17, 40, 49
43, 50, 74	21, 53, 62	28, 31, 68	22, 69, 76	12, 34, 48
34, 72, 76	17, 32, 66	19, 37, 78	43, 67, 81	4, 37, 81
1, 60, 64	5, 17, 83	42, 65, 72	3, 42, 84	2, 27, 29
24, 57, 82	8, 74, 82	35, 49, 59	1, 8, 34	7, 53, 80
53, 63, 84	27, 42, 49	21, 33, 45	16, 60, 66	9, 51, 69
18, 30, 63	16, 22, 25	44, 69, 72	6, 27, 78	12, 45, 69
48, 63, 74	4, 24, 76	17, 41, 57	11, 30, 81	33, 48, 58
3, 35, 38	8, 43, 68	2, 31, 78	37, 58, 73	30, 36, 46
10, 16, 28	25, 28, 72	13, 75, 80	15, 33, 76	14, 55, 60
43, 53, 72	8, 60, 75	5, 45, 57	5, 15, 71	6, 17, 39
6, 63, 73	22, 79, 82	37, 56, 67	26, 71, 75	21, 67, 76
49, 79, 81	21, 52, 74	9, 49, 78	2, 42, 60	9, 44, 71
27, 43, 46	5, 48, 55	15, 22, 67	7, 26, 31	38, 59, 62

27, 57, 81	19, 28, 49	21, 42, 48	20, 31, 46	17, 50, 71
39, 54, 64	12, 32, 44	34, 37, 78	38, 53, 57	5, 18, 27
1, 44, 75	1, 24, 43	8, 67, 75	19, 22, 46	1, 59, 67
27, 50, 83	48, 68, 77	21, 44, 60	51, 79, 83	18, 66, 73
65, 71, 76	41, 68, 78	0, 20, 56	12, 22, 25	6, 29, 78
24, 53, 78	43, 48, 78	16, 39, 64	15, 45, 51	20, 70, 76
9, 73, 81	28, 71, 76	12, 49, 58	16, 58, 73	31, 34, 49
59, 65, 84	0, 25, 35	29, 47, 84	55, 69, 76	33, 67, 76
1, 25, 83	19, 62, 74	46, 57, 64	7, 58, 67	6, 15, 57
42, 53, 74	1, 27, 61	4, 14, 77	13, 21, 49	4, 45, 73
41, 80, 82	10, 34, 84	0, 44, 60	11, 46, 61	68, 70, 73
4, 64, 75	15, 41, 74	13, 27, 80	39, 52, 83	4, 9, 40
14, 33, 71	12, 57, 69	19, 26, 72	10, 32, 47	10, 19, 32
0, 32, 55	29, 39, 48	41, 63, 70	26, 31, 53	21, 45, 69
6, 19, 47	24, 48, 75	7, 22, 40	9, 37, 42	68, 74, 80
44, 51, 78	10, 45, 77	34, 40, 79	23, 29, 56	22, 47, 66
30, 34, 41	3, 8, 50	33, 52, 60	12, 33, 72	29, 31, 42
2, 56, 65	25, 63, 78	20, 44, 65	38, 47, 50	43, 46, 61
11, 34, 70	0, 40, 54	40, 43, 52	38, 44, 53	15, 62, 75
26, 35, 83	30, 63, 78	13, 58, 60	33, 66, 75	57, 63, 66
0, 37, 42	27, 64, 84	6, 50, 77	13, 50, 71	18, 23, 41
33, 47, 77	38, 49, 68	5, 48, 61	22, 24, 42	19, 31, 38
28, 34, 65	25, 51, 58	30, 67, 69	9, 43, 55	46, 70, 81
24, 39, 83	6, 17, 44	58, 63, 71	0, 8, 58	41, 45, 72
23, 25, 68	40, 42, 61	22, 30, 48	4, 6, 39	6, 64, 83
20, 28, 50	15, 25, 27	12, 82, 84	22, 26, 74	26, 44, 66

24, 42, 76	46, 59, 77	0, 2, 18	51, 71, 82	16, 64, 67
70, 82, 84	9, 57, 83	2, 48, 51	0, 25, 76	54, 74, 76
11, 42, 62	3, 49, 77	18, 20, 49	24, 61, 82	29, 55, 84
2, 22, 37	21, 28, 72	3, 47, 61	24, 58, 60	17, 21, 26
0, 35, 69	2, 61, 83	65, 72, 79	47, 52, 83	22, 45, 71
6, 9, 36	17, 38, 67	31, 52, 55	9, 49, 59	11, 26, 29
20, 65, 71	32, 36, 65	32, 44, 83	17, 44, 73	2, 48, 81
6, 14, 53	18, 55, 64	2, 41, 56	21, 36, 77	3, 45, 78
59, 72, 84	5, 50, 68	28, 49, 71	20, 29, 72	26, 59, 69
6, 31, 50	56, 81, 84	4, 41, 80	20, 53, 82	14, 18, 59
10, 14, 75	40, 63, 73	10, 33, 53	27, 32, 84	17, 35, 81
0, 61, 81	15, 30, 75	20, 43, 76	1, 32, 84	12, 47, 53
38, 40, 82	8, 70, 77	4, 71, 83	31, 49, 75	2, 8, 40
14, 72, 83	18, 52, 77	15, 29, 59	26, 48, 82	33, 73, 79
11, 25, 66	8, 22, 44	8, 44, 64	33, 53, 67	13, 15, 72
12, 73, 76	5, 9, 39	6, 37, 66	2, 9, 36	8, 42, 79
53, 66, 76	47, 62, 84	6, 30, 60	3, 65, 68	24, 35, 49
14, 20, 48	8, 31, 55	17, 41, 83	0, 10, 63	1, 70, 76
14, 20, 52	14, 23, 45	6, 22, 61	1, 50, 56	9, 18, 78
2, 22, 26	12, 24, 48	3, 23, 33	46, 56, 64	3, 7, 62
53, 64, 78	17, 30, 59	12, 45, 53	15, 18, 46	35, 56, 75
6, 10, 35	27, 35, 38	47, 50, 83	25, 37, 76	7, 30, 50
25, 39, 84	8, 31, 62	27, 50, 63	1, 60, 80	48, 59, 71
26, 41, 47	46, 49, 51	12, 50, 74	6, 11, 33	15, 28, 65
36, 45, 79	19, 61, 71	38, 58, 81	2, 45, 68	21, 46, 81
3, 62, 83	8, 10, 36	0, 61, 64	34, 62, 77	23, 74, 83

6, 25, 40	33, 48, 63	58, 69, 80	13, 16, 53	33, 52, 62
15, 73, 80	7, 56, 78	20, 24, 47	3, 14, 16	37, 47, 57
2, 23, 37	23, 26, 50	13, 21, 66	26, 35, 76	2, 50, 84
10, 16, 37	14, 45, 58	14, 18, 51	4, 44, 75	8, 59, 73
28, 81, 83	39, 43, 67	21, 46, 74	20, 40, 55	6, 59, 80
6, 62, 72	25, 29, 44	11, 52, 55	42, 52, 64	19, 34, 55
15, 47, 76	28, 47, 69	13, 59, 62	6, 34, 76	4, 44, 50
1, 4, 54	12, 16, 74	16, 48, 74	8, 52, 84	21, 60, 84
8, 31, 39	7, 74, 81	25, 30, 80	5, 76, 84	0, 49, 60
21, 25, 53	8, 15, 37	47, 58, 71	24, 46, 76	27, 38, 63
30, 40, 61	37, 69, 80	40, 46, 75	25, 67, 83	22, 28, 33
7, 25, 71	26, 45, 50	25, 29, 70	18, 20, 64	22, 35, 68
3, 58, 68	21, 32, 75	1, 43, 45	8, 10, 23	4, 53, 77
14, 41, 79	11, 33, 35	44, 51, 80	44, 46, 84	19, 44, 75
27, 30, 67	34, 63, 75	10, 16, 17	19, 50, 78	21, 52, 59
51, 73, 76	22, 49, 69	11, 26, 84	56, 59, 83	4, 15, 70
15, 37, 82	27, 31, 41	9, 34, 48	56, 69, 76	7, 35, 39
12, 37, 49	20, 56, 79	21, 70, 78	0, 22, 34	52, 54, 67
0, 19, 84	25, 41, 73	6, 48, 71	35, 40, 47	33, 45, 63
7, 26, 30	18, 80, 83	9, 40, 67	34, 53, 77	28, 42, 57
2, 15, 38	0, 48, 70	24, 27, 64	0, 14, 22	11, 49, 72
0, 21, 30	45, 51, 55	43, 72, 81	16, 46, 80	20, 39, 58
24, 49, 57	34, 68, 82	5, 24, 66	18, 31, 81	40, 50, 63
9, 28, 51	16, 34, 73	7, 79, 82	25, 79, 84	45, 59, 70
3, 38, 47	36, 60, 70	5, 23, 27	1, 4, 22	4, 11, 73
26, 73, 80	27, 34, 60	3, 18, 76	16, 44, 65	19, 76, 80

51, 54, 70	27, 56, 60	25, 48, 71	8, 51, 74	45, 58, 66
8, 30, 35	49, 53, 55	5, 14, 84	47, 56, 62	23, 38, 82
47, 59, 73	18, 42, 79	55, 67, 73	13, 48, 61	0, 28, 56
32, 73, 77	16, 17, 75	28, 46, 50	22, 58, 72	10, 67, 68
38, 43, 79	6, 8, 53	14, 34, 37	51, 63, 76	15, 24, 45
25, 40, 74	14, 47, 70	18, 32, 72	53, 58, 62	14, 47, 74
20, 30, 46	54, 76, 79	38, 60, 71	36, 39, 45	4, 29, 53
1, 38, 63	10, 44, 68	11, 49, 70	5, 46, 49	3, 30, 77
4, 17, 46	33, 47, 54	8, 13, 42	27, 65, 71	1, 22, 72
11, 23, 34	7, 15, 29	23, 25, 57	7, 42, 56	4, 26, 29
4, 13, 49	37, 49, 64	8, 17, 83	8, 33, 49	7, 18, 25
28, 54, 74	2, 30, 43	51, 63, 71	40, 49, 62	23, 79, 84
4, 25, 51	22, 71, 80	23, 37, 41	3, 79, 82	46, 58, 68
31, 52, 63	4, 8, 26	29, 31, 44	27, 57, 73	23, 73, 84
9, 59, 64	4, 48, 79	13, 35, 64	37, 62, 75	35, 76, 84
3, 19, 54	61, 67, 77	4, 19, 60	6, 38, 58	22, 45, 53
6, 29, 71	14, 50, 76	13, 19, 24	29, 55, 59	5, 59, 75
25, 41, 51	23, 46, 60	35, 66, 72	4, 42, 68	4, 32, 52
57, 73, 79	8, 45, 66	37, 52, 77	3, 61, 64	23, 30, 72
1, 34, 71	2, 29, 47	28, 47, 75	6, 59, 72	8, 32, 60
0, 27, 74	6, 66, 79	39, 47, 60	18, 25, 36	10, 42, 55
36, 45, 61	21, 35, 66	12, 23, 31	4, 26, 34	30, 49, 55
4, 49, 74	9, 63, 72	10, 30, 54	0, 5, 72	2, 49, 63
23, 29, 81	24, 51, 53	21, 57, 80	5, 16, 60	9, 34, 57
4, 13, 80	18, 26, 33	34, 48, 69	45, 48, 83	35, 73, 83
38, 45, 64	11, 22, 57	3, 25, 70	9, 64, 75	6, 57, 60

17, 29, 56	33, 36, 56	18, 43, 57	1, 17, 47	2, 45, 75
13, 36, 69	24, 32, 55	13, 18, 53	27, 39, 55	5, 50, 64
36, 43, 59	2, 10, 12	3, 9, 48	22, 63, 81	14, 40, 71
3, 14, 81	2, 23, 65	18, 29, 65	18, 40, 45	36, 58, 76
5, 38, 80	13, 57, 60	7, 41, 66	44, 50, 81	37, 52, 80
36, 50, 58	26, 50, 55	13, 22, 76	41, 53, 64	12, 18, 26
25, 32, 70	32, 63, 82	2, 39, 72	4, 16, 47	67, 74, 76
26, 33, 78	20, 41, 64	24, 40, 59	20, 36, 60	26, 32, 71
3, 39, 46	44, 46, 58	21, 65, 76	22, 25, 53	19, 33, 69
10, 19, 58	18, 45, 70	22, 30, 42	23, 30, 44	30, 78, 81
8, 43, 65	5, 57, 80	24, 52, 80	15, 42, 77	18, 58, 81
5, 47, 84	26, 44, 77	20, 49, 77	28, 77, 83	17, 39, 53
44, 49, 53	9, 43, 46	15, 45, 59	20, 50, 72	10, 69, 74
3, 64, 79	18, 60, 78	10, 15, 41	0, 8, 41	6, 16, 45
4, 8, 27	0, 9, 67	23, 53, 66	11, 13, 74	17, 20, 84
31, 74, 84	19, 59, 74	50, 51, 84	10, 17, 75	3, 10, 55
44, 46, 84	40, 69, 79	6, 31, 49	3, 6, 80	9, 54, 59
34, 53, 68	6, 21, 67	28, 53, 84	44, 52, 57	33, 52, 65
7, 17, 63	2, 38, 72	32, 43, 64	13, 25, 58	1, 7, 9
25, 40, 81	32, 50, 59	40, 44, 54	11, 39, 81	15, 20, 41
19, 38, 67	17, 22, 37	5, 20, 62	33, 59, 81	21, 33, 78
18, 48, 60	31, 42, 54	7, 21, 65	32, 37, 62	38, 58, 61
32, 43, 71	6, 48, 62	21, 49, 65	16, 30, 41	17, 28, 58
13, 23, 83	4, 28, 42	25, 52, 56	4, 56, 74	5, 43, 56
45, 55, 65	12, 31, 33	10, 46, 48	61, 69, 72	7, 13, 82
29, 63, 74	67, 70, 77	72, 77, 84	49, 52, 59	15, 73, 76

31, 61, 76	7, 25, 78	0, 19, 45	37, 52, 76	8, 22, 57
5, 47, 68	35, 48, 54	8, 20, 69	31, 35, 73	4, 48, 79
14, 17, 44	44, 55, 83	24, 38, 65	23, 42, 46	35, 44, 77
9, 40, 71	37, 78, 84	30, 51, 62	38, 63, 65	2, 5, 59
19, 54, 60	15, 20, 25	3, 38, 75	16, 17, 82	46, 75, 84
8, 30, 45	55, 63, 77	0, 13, 32	24, 49, 54	50, 74, 80
0, 47, 54	0, 13, 37	47, 49, 76	11, 44, 51	46, 66, 77
5, 32, 40	10, 26, 46	31, 56, 68	5, 42, 71	9, 14, 27
37, 43, 84	20, 38, 81	49, 54, 69	3, 60, 69	19, 35, 55
9, 24, 72	53, 56, 71	22, 38, 74	30, 52, 65	22, 59, 77
6, 14, 69	54, 75, 84	12, 51, 56	42, 54, 78	38, 50, 73
1, 37, 41	16, 49, 55	0, 45, 47	30, 65, 83	17, 28, 30
50, 53, 55	7, 19, 65	13, 36, 44	36, 47, 52	31, 50, 83
7, 61, 69	11, 46, 72	62, 68, 72	9, 71, 82	2, 16, 29
8, 10, 51	13, 15, 49	33, 44, 69	6, 28, 58	10, 40, 76
14, 24, 60	29, 41, 51	7, 77, 81	8, 47, 51	30, 49, 51
16, 30, 61	31, 39, 43	0, 18, 23	29, 32, 37	18, 36, 40
52, 68, 83	29, 35, 45	14, 65, 74	16, 27, 78	26, 38, 65
6, 42, 61	17, 30, 76	10, 76, 81	56, 72, 84	5, 35, 55
28, 40, 42	1, 38, 60	17, 40, 59	13, 23, 63	12, 37, 63
42, 47, 60	30, 69, 83	22, 46, 56	6, 46, 55	9, 35, 65
24, 27, 56	35, 40, 58	17, 27, 52	3, 44, 83	6, 48, 63
3, 5, 70	57, 59, 69	26, 31, 67	52, 58, 60	75, 81, 83
18, 48, 56	19, 72, 74	7, 55, 66	25, 49, 75	22, 32, 74
0, 6, 70	12, 16, 58	56, 60, 78	48, 55, 77	2, 15, 36
33, 37, 70	42, 64, 69	16, 21, 63	46, 50, 80	58, 74, 83

66, 69, 74	37, 57, 64	50, 64, 69	3, 13, 35	19, 35, 60
20, 70, 82	4, 16, 48	1, 19, 25	27, 41, 75	13, 51, 69
9, 13, 41	4, 34, 46	25, 48, 75	33, 34, 40	27, 54, 83
2, 59, 78	43, 68, 75	50, 51, 77	23, 41, 71	2, 20, 23
10, 31, 62	22, 52, 64	11, 69, 82	11, 18, 74	26, 70, 81
17, 57, 77	23, 48, 51	7, 22, 45	41, 48, 81	26, 39, 59
34, 47, 79	14, 18, 29	61, 69, 71	19, 25, 63	13, 37, 50
31, 40, 45	6, 13, 42	22, 34, 72	7, 14, 76	38, 69, 73
39, 75, 82	25, 55, 62	6, 65, 76	5, 23, 48	15, 55, 68
14, 21, 70	38, 59, 69	29, 47, 82	38, 52, 54	54, 58, 80
24, 37, 63	35, 50, 77	23, 36, 50	2, 47, 53	10, 50, 61
20, 34, 54	3, 5, 7	41, 43, 82	5, 40, 53	44, 53, 67
3, 34, 43	26, 57, 81	51, 55, 69	15, 40, 65	12, 36, 67
21, 25, 60	8, 26, 60	29, 42, 45	1, 36, 75	42, 47, 50
8, 27, 37	12, 55, 76	61, 67, 79	33, 34, 47	2, 32, 46
1, 55, 58	16, 23, 76	27, 33, 79	26, 48, 84	12, 15, 37
35, 60, 63	3, 22, 32	0, 2, 11	11, 48, 67	14, 19, 79
4, 67, 82	6, 9, 82	45, 55, 61	24, 29, 84	25, 54, 79
21, 29, 71	25, 27, 82	3, 34, 84	43, 62, 75	24, 70, 78
4, 71, 77	4, 12, 67	0, 7, 38	36, 63, 81	3, 25, 54
10, 44, 47	20, 23, 79	43, 51, 81	14, 42, 67	50, 52, 82
4, 30, 40	11, 31, 40	0, 7, 59	10, 19, 43	13, 77, 82
31, 41, 77	19, 62, 77	1, 11, 62	15, 40, 46	10, 49, 71
42, 44, 56	5, 36, 54	3, 15, 29	0, 9, 66	11, 58, 84
31, 46, 50	37, 56, 69	4, 20, 61	27, 45, 82	12, 34, 46
46, 54, 65	4, 65, 69	24, 62, 77	9, 21, 32	20, 52, 76

9, 56, 72	21, 43, 54	41, 50, 65	5, 10, 30	17, 60, 63
11, 27, 33	9, 43, 66	3, 15, 52	16, 46, 69	13, 56, 67
2, 9, 73	2, 58, 79	0, 63, 65	22, 24, 70	5, 8, 19
12, 14, 66	38, 42, 44	40, 73, 83	7, 30, 32	17, 36, 51
55, 63, 79	37, 53, 65	25, 50, 64	16, 20, 27	16, 53, 61
14, 33, 54	4, 43, 76	13, 34, 43	10, 34, 62	13, 62, 69
5, 63, 76	9, 33, 80	36, 44, 56	4, 7, 17	41, 70, 83
1, 56, 81	49, 60, 69	57, 61, 84	33, 39, 42	1, 10, 21
30, 43, 67	16, 31, 40	15, 36, 66	59, 61, 77	46, 55, 57
34, 36, 76	8, 73, 75	9, 21, 23	65, 69, 84	59, 68, 72
24, 71, 84	11, 54, 82	21, 37, 73	3, 21, 81	54, 56, 83
11, 22, 38	29, 38, 57	60, 79, 81	9, 60, 74	50, 53, 56
24, 27, 77	18, 41, 73	8, 11, 55	1, 36, 51	12, 17, 35
31, 51, 77	12, 57, 71	47, 61, 77	10, 18, 32	25, 39, 61
31, 77, 82	3, 8, 36	3, 15, 63	0, 24, 42	13, 65, 68
9, 28, 53	5, 59, 65	7, 28, 80	41, 53, 76	2, 16, 59
22, 66, 82	13, 24, 67	8, 13, 38	24, 73, 81	16, 22, 65
46, 74, 78	9, 15, 69	28, 33, 41	35, 42, 63	5, 51, 75
5, 37, 61	2, 18, 46	7, 40, 68	49, 67, 80	6, 49, 67
1, 6, 51	54, 59, 61	38, 51, 64	56, 65, 77	50, 62, 76
16, 20, 57	9, 14, 19	54, 77, 83	2, 68, 73	59, 74, 84
51, 54, 60	1, 3, 24	3, 51, 65	21, 28, 82	20, 29, 68
13, 34, 70	13, 29, 43	23, 39, 77	37, 44, 63	0, 24, 74
39, 41, 74	12, 23, 59	22, 29, 40	9, 25, 46	37, 69, 75
26, 57, 63	11, 32, 50	16, 34, 41	17, 40, 62	22, 29, 78
33, 68, 75	8, 41, 67	31, 64, 72	6, 24, 46	4, 43, 83

10, 22, 83	16, 71, 81	26, 45, 77	8, 29, 34	7, 69, 74
45, 74, 83	15, 17, 81	21, 32, 67	14, 57, 73	18, 55, 74
6, 28, 35	25, 28, 72	7, 43, 70	10, 36, 68	14, 35, 63
59, 65, 67	9, 27, 47	16, 23, 62	27, 51, 61	10, 24, 37
60, 62, 65	3, 9, 56	9, 42, 80	1, 11, 51	7, 34, 45
6, 16, 36	5, 12, 65	34, 42, 45	13, 42, 78	16, 18, 81
34, 37, 44	9, 19, 44	0, 38, 77	12, 40, 63	35, 43, 49
30, 54, 62	5, 21, 31	34, 38, 41	16, 25, 43	22, 27, 31
3, 47, 60	3, 34, 64	9, 50, 65	14, 40, 65	2, 35, 76
5, 39, 62	63, 75, 79	24, 47, 73	54, 68, 77	9, 15, 70
14, 51, 66	32, 38, 70	15, 78, 84	8, 29, 50	27, 34, 67
62, 67, 80	54, 57, 68	13, 19, 46	43, 56, 80	14, 36, 56
10, 25, 82	40, 66, 81	3, 37, 74	37, 65, 74	1, 15, 20
1, 29, 80	14, 21, 25	6, 65, 84	14, 39, 75	10, 26, 35
23, 43, 80	10, 23, 56	1, 50, 52	20, 58, 61	26, 56, 79
2, 18, 38	44, 47, 78	14, 26, 83	48, 50, 72	48, 77, 84
7, 36, 78	0, 2, 11	0, 6, 51	20, 26, 75	27, 30, 76
17, 48, 54	4, 25, 49	4, 45, 63	4, 30, 38	11, 23, 77
75, 77, 79	0, 44, 52	6, 17, 53	0, 23, 61	0, 5, 77
2, 41, 66	7, 56, 84	15, 28, 79	13, 34, 83	14, 60, 73
35, 51, 70	2, 55, 84	13, 39, 48	9, 11, 50	16, 32, 56
17, 24, 29	1, 52, 77	23, 49, 79	13, 45, 52	9, 38, 79
19, 23, 26	52, 67, 82	16, 34, 82	9, 31, 81	3, 52, 73
54, 56, 70	12, 43, 78	12, 33, 77	0, 26, 58	15, 24, 81
52, 63, 72	16, 44, 62	3, 12, 75	4, 24, 82	23, 67, 71
23, 28, 62	11, 31, 60	15, 38, 48	37, 40, 51	24, 58, 77

27, 29, 69	2, 49, 82	19, 41, 52	51, 54, 64	38, 57, 72
2, 7, 21	31, 36, 59	19, 45, 70	0, 71, 74	29, 49, 60
10, 18, 63	27, 36, 49	40, 43, 65	17, 47, 81	26, 40, 48
43, 48, 83	62, 69, 81	0, 10, 15	9, 34, 81	19, 27, 68
16, 38, 49	48, 53, 59	3, 27, 80	38, 60, 62	48, 54, 73
51, 53, 72	13, 33, 44	4, 6, 70	28, 46, 77	36, 69, 80
14, 23, 37	20, 51, 83	28, 41, 74	3, 11, 59	4, 35, 45
23, 47, 54	63, 67, 83	26, 51, 66	12, 54, 62	0, 10, 29
11, 48, 60	47, 49, 57	28, 32, 61	2, 15, 63	32, 39, 81
17, 43, 77	9, 29, 76	15, 56, 82	11, 17, 76	24, 51, 71
28, 40, 68	16, 37, 39	34, 64, 79	7, 46, 73	1, 28, 65
17, 21, 24	11, 41, 69	9, 11, 74	23, 49, 63	9, 39, 49
11, 18, 61	6, 58, 63	6, 40, 82	10, 51, 56	0, 65, 79
38, 74, 84	42, 48, 67	10, 13, 57	11, 19, 43	28, 63, 84
52, 61, 63	18, 24, 28	15, 23, 81	30, 36, 74	2, 32, 51
13, 31, 73	42, 51, 72	19, 57, 84	22, 44, 64	5, 21, 74
1, 44, 66	5, 28, 54	6, 29, 69	3, 8, 69	38, 44, 48
37, 57, 67	28, 30, 84	20, 58, 83	5, 24, 33	15, 18, 79
40, 53, 77	66, 82, 84	36, 50, 78	1, 7, 67	21, 63, 81
32, 42, 81	25, 36, 57	11, 26, 71	3, 53, 74	19, 22, 48
46, 65, 81	8, 22, 77	5, 14, 62	7, 23, 70	15, 48, 64
7, 27, 73	2, 25, 74	0, 57, 79	11, 37, 71	68, 72, 82
8, 47, 62	9, 29, 52	24, 28, 57	60, 72, 81	33, 70, 80
21, 31, 61	19, 57, 81	18, 21, 36	3, 11, 18	17, 36, 66
9, 23, 35	4, 27, 46	22, 27, 79	64, 71, 78	7, 28, 80
22, 57, 70	0, 7, 64	1, 46, 72	10, 36, 65	58, 72, 80

53, 75, 82	33, 61, 68	16, 54, 73	29, 81, 83	25, 44, 55
1, 77, 80	12, 39, 77	0, 8, 11	7, 32, 35	39, 41, 48
6, 9, 68	0, 37, 46	3, 11, 83	39, 55, 84	45, 59, 79
6, 13, 18	0, 48, 50	12, 28, 82	7, 12, 57	15, 26, 60
5, 52, 72	3, 44, 82	46, 71, 73	35, 47, 65	34, 45, 74
41, 49, 54	2, 24, 50	33, 76, 82	4, 20, 80	39, 42, 66
14, 28, 55	5, 29, 67	16, 47, 72	12, 30, 64	64, 70, 73
4, 18, 62	12, 62, 74	28, 39, 71	0, 3, 73	17, 66, 71
24, 65, 70	22, 55, 81	29, 38, 49	2, 69, 81	11, 59, 80
20, 38, 45	1, 23, 50	37, 46, 70	31, 76, 78	30, 37, 59
38, 70, 72	32, 42, 57	8, 19, 26	21, 47, 63	20, 28, 35
5, 37, 43	37, 50, 82	41, 55, 59	43, 48, 83	15, 54, 72
41, 44, 65	2, 80, 84	8, 12, 68	12, 58, 64	11, 14, 38
18, 43, 82	25, 41, 43	9, 41, 84	24, 50, 54	31, 55, 80
19, 54, 82	32, 57, 79	13, 38, 66	8, 40, 84	2, 55, 69
18, 33, 70	2, 54, 65	31, 45, 81	10, 48, 70	14, 35, 47
12, 42, 63	1, 48, 76	2, 35, 55	1, 28, 35	4, 14, 17
14, 29, 77	39, 42, 82	26, 36, 73	16, 19, 66	29, 60, 68
10, 59, 83	13, 40, 76	40, 45, 77	15, 71, 77	60, 80, 83
39, 53, 79	33, 65, 79	4, 54, 66	54, 56, 75	4, 7, 51
10, 35, 80	0, 43, 69	3, 50, 52	2, 5, 31	23, 28, 32
1, 20, 78	50, 61, 75	1, 6, 83	27, 55, 75	36, 55, 63
11, 46, 59	9, 16, 83	4, 23, 78	16, 19, 51	12, 28, 48
18, 21, 54	12, 42, 65	53, 56, 62	40, 54, 82	29, 45, 65
17, 52, 84	64, 68, 77	25, 57, 66	1, 30, 77	11, 44, 76
1, 32, 78	55, 68, 71	7, 55, 64	16, 31, 57	1, 10, 33

63, 74, 77	21, 26, 82	23, 32, 59	16, 29, 75	9, 63, 82
8, 19, 79	6, 12, 55	30, 71, 81	13, 31, 71	8, 18, 61
12, 53, 65	23, 28, 69	6, 19, 37	1, 17, 61	6, 22, 75
0, 39, 78	8, 14, 69	3, 22, 61	27, 37, 62	0, 12, 21
10, 13, 38	1, 6, 62	25, 27, 59	1, 57, 68	4, 36, 52
34, 52, 70	14, 16, 65	16, 19, 83	66, 74, 78	30, 33, 80
67, 74, 81	4, 61, 82	55, 62, 81	13, 26, 65	36, 39, 44
51, 65, 70	0, 80, 82	9, 44, 63	12, 41, 68	1, 53, 64
55, 77, 79	20, 28, 55	32, 53, 75	38, 45, 56	2, 45, 80
5, 18, 67	48, 78, 82	34, 52, 66	11, 31, 56	2, 12, 75
8, 14, 77	12, 27, 40	7, 40, 60	32, 50, 79	33, 43, 64
71, 75, 78	4, 69, 84	47, 59, 63	27, 39, 70	6, 43, 71
7, 26, 49	16, 27, 47	0, 14, 24	14, 28, 74	17, 56, 78
26, 52, 56	10, 26, 64	17, 31, 68	0, 12, 83	14, 39, 76
17, 22, 76	0, 32, 36	5, 44, 67	40, 44, 80	63, 69, 77
21, 55, 83	8, 50, 65	5, 11, 77	7, 36, 38	16, 71, 77
31, 36, 83	31, 63, 69	24, 31, 69	7, 29, 72	24, 26, 44
22, 62, 69	23, 70, 75	16, 35, 51	41, 75, 79	2, 39, 53
29, 53, 58	12, 19, 64	36, 57, 71	11, 20, 65	4, 9, 20
42, 73, 81	7, 47, 74	3, 26, 74	34, 38, 64	32, 41, 71
55, 58, 66	16, 28, 43	49, 80, 83	1, 76, 81	9, 32, 62
6, 24, 43	5, 28, 63	0, 76, 81	47, 56, 80	24, 38, 79
39, 58, 84	30, 57, 72	17, 42, 46	34, 67, 75	2, 10, 52
40, 44, 70	15, 30, 64	3, 31, 80	25, 48, 52	3, 63, 72
37, 70, 80	3, 13, 41	11, 64, 66	24, 30, 80	66, 78, 81
15, 34, 44	22, 43, 79	66, 68, 76	18, 71, 80	26, 30, 78

1, 12, 73	33, 58, 82	29, 43, 62	31, 43, 57	1, 40, 72
62, 65, 70	19, 35, 70	11, 53, 63	15, 31, 47	23, 44, 74
19, 71, 83	38, 43, 77	2, 34, 67	19, 33, 37	27, 32, 40
1, 16, 35	25, 53, 69	12, 50, 70	4, 23, 52	17, 31, 38
14, 48, 50	22, 47, 67	26, 47, 82	27, 39, 69	16, 18, 77
4, 15, 57	7, 18, 50	39, 44, 63	5, 9, 24	56, 71, 82
12, 14, 84	17, 22, 84	11, 30, 57	55, 61, 79	6, 26, 73
59, 79, 83	30, 51, 79	0, 41, 52	41, 62, 81	4, 29, 34
23, 43, 47	11, 37, 68	27, 35, 75	36, 48, 80	13, 20, 54
3, 31, 42	31, 35, 70	5, 32, 65	11, 22, 52	36, 61, 72
6, 51, 80	44, 49, 72	23, 29, 54	8, 54, 81	45, 49, 63
18, 66, 80	6, 33, 73	57, 63, 76	11, 45, 84	6, 43, 59
36, 58, 78	1, 9, 30	9, 47, 74	64, 77, 80	21, 59, 75
15, 21, 78	40, 75, 83	7, 70, 83	57, 67, 68	12, 23, 51
8, 32, 65	3, 44, 77	18, 55, 68	14, 41, 81	4, 12, 50
17, 70, 79	17, 68, 80	30, 35, 84	41, 73, 84	0, 26, 42
11, 35, 56	41, 59, 61	20, 39, 45	11, 32, 56	19, 67, 69
45, 73, 75	8, 44, 80	28, 59, 64	22, 41, 54	29, 54, 64
25, 37, 77	23, 60, 73	30, 52, 70	0, 28, 53	12, 15, 26
5, 16, 63	3, 10, 58	2, 67, 80	9, 12, 62	39, 47, 52
0, 71, 73	36, 39, 46	14, 43, 54	5, 8, 81	41, 44, 57
9, 18, 76	37, 74, 79	7, 39, 46	26, 55, 58	38, 40, 46
33, 71, 84	6, 68, 82	36, 54, 75	38, 54, 74	15, 57, 75
22, 49, 81	52, 55, 73	3, 18, 69	46, 74, 79	15, 42, 69
3, 36, 57	19, 39, 51	62, 64, 73	21, 53, 61	32, 34, 78
45, 52, 67	39, 50, 54	29, 63, 70	22, 41, 60	5, 10, 14

36, 74, 77	34, 54, 84	10, 42, 71	15, 40, 68	6, 45, 47
3, 29, 32	5, 10, 42	35, 46, 71	42, 46, 58	13, 47, 78
8, 17, 63	11, 20, 57	1, 39, 63	12, 18, 40	23, 68, 73
33, 36, 46	25, 43, 64	3, 57, 76	18, 53, 73	28, 44, 68
1, 30, 40	49, 64, 67	5, 15, 43	35, 74, 81	2, 69, 83
0, 29, 36	20, 61, 73	15, 38, 61	1, 12, 79	15, 21, 39
55, 65, 78	49, 53, 73	31, 53, 74	23, 64, 67	42, 47, 66
16, 40, 61	10, 21, 34	14, 23, 52	15, 25, 34	4, 23, 39
35, 39, 57	21, 28, 52	7, 66, 69	16, 54, 67	61, 63, 66
49, 58, 70	7, 16, 44	1, 15, 62	20, 44, 48	1, 13, 71
15, 52, 56	20, 22, 80	22, 33, 43	6, 45, 68	8, 15, 72
3, 40, 70	5, 29, 77	47, 61, 70	34, 54, 66	19, 21, 57
25, 38, 75	46, 49, 70	17, 45, 65	20, 51, 79	21, 40, 80
5, 45, 56	11, 51, 78	29, 36, 49	31, 47, 51	10, 63, 71
38, 73, 78	16, 57, 59	20, 45, 78	10, 21, 56	13, 46, 65
1, 5, 82	3, 27, 43	3, 14, 67	7, 16, 54	11, 21, 54
8, 13, 28	2, 60, 83	10, 15, 61	10, 66, 82	5, 17, 52
2, 17, 50	7, 62, 64	4, 59, 68	32, 60, 80	10, 66, 81
3, 21, 82	16, 69, 83	7, 10, 83	10, 39, 50	13, 17, 74
31, 61, 74	1, 21, 55	55, 73, 78	4, 41, 43	2, 12, 43
9, 30, 37	3, 17, 26	7, 19, 31	28, 40, 75	17, 25, 56
14, 25, 45	8, 17, 74	20, 27, 78	36, 42, 76	26, 40, 62
15, 48, 82	10, 39, 57	2, 22, 27	12, 19, 70	16, 40, 60
15, 53, 80	19, 24, 55	11, 42, 55	16, 45, 56	19, 28, 73
5, 20, 83	4, 18, 59	0, 62, 82	39, 51, 66	9, 12, 61
19, 40, 67	32, 59, 73	13, 18, 39	13, 20, 26	7, 47, 73

9, 22, 52	29, 51, 74	26, 67, 72	55, 71, 75	29, 36, 66
28, 63, 70	34, 40, 74	0, 65, 75	49, 52, 84	18, 24, 37
41, 46, 75	22, 30, 73	14, 32, 61	27, 38, 41	31, 56, 82
0, 33, 43	38, 43, 80	16, 41, 50	1, 33, 82	13, 28, 32
45, 64, 76	48, 67, 72	32, 48, 58	67, 70, 74	29, 32, 39
34, 58, 61	0, 48, 51	7, 16, 79	9, 30, 45	34, 49, 56
0, 62, 83	12, 20, 66	28, 43, 61	21, 27, 58	19, 40, 64
18, 30, 74	72, 76, 79	19, 42, 50	4, 31, 78	46, 60, 82
7, 15, 49	22, 43, 58	19, 52, 76	25, 63, 65	16, 28, 70
6, 10, 12	5, 11, 53	16, 62, 78	17, 21, 64	2, 16, 52
4, 64, 74	45, 47, 75	0, 4, 19	24, 33, 55	9, 61, 75
0, 13, 54	53, 65, 83	29, 35, 57	58, 78, 83	2, 8, 71
3, 48, 66	2, 56, 63	3, 51, 65	28, 48, 52	14, 36, 43
2, 26, 57	3, 35, 66	10, 23, 45	10, 49, 72	1, 37, 81
18, 47, 69	0, 28, 39	26, 60, 78	17, 69, 82	22, 71, 78
3, 27, 51	26, 61, 84	38, 56, 77	34, 55, 80	26, 32, 54
14, 57, 61	29, 48, 60	45, 80, 84	8, 63, 68	30, 62, 72
26, 49, 62	23, 58, 82	8, 24, 33	60, 67, 75	4, 19, 39
33, 57, 83	0, 36, 49	56, 58, 65	5, 13, 17	29, 61, 65
4, 24, 62	15, 66, 84	9, 29, 68	54, 63, 84	6, 52, 64
19, 58, 79	27, 30, 45	29, 33, 57	13, 23, 35	5, 56, 71
20, 51, 73	9, 48, 84	24, 72, 79	19, 34, 56	21, 41, 51
10, 52, 57	13, 40, 72	32, 46, 68	22, 50, 58	9, 31, 35
27, 37, 53	22, 62, 68	2, 4, 61	30, 44, 84	2, 66, 77
18, 66, 78	24, 63, 67	13, 64, 70	8, 30, 49	2, 30, 32
11, 36, 72	33, 40, 51	56, 64, 66	2, 21, 34	1, 20, 31

14, 27, 46	37, 50, 81	40, 47, 67	25, 37, 59	3, 16, 24
2, 47, 58	21, 27, 48	18, 49, 65	35, 37, 45	15, 63, 69
9, 20, 33	5, 8, 81	11, 25, 61	14, 24, 44	20, 63, 82
27, 52, 58	13, 67, 77	17, 29, 62	3, 31, 45	25, 36, 81
27, 48, 76	49, 61, 83	2, 54, 78	20, 32, 40	13, 26, 62
3, 33, 49	2, 7, 57	16, 24, 67	22, 53, 80	21, 27, 62
35, 60, 83	24, 32, 64	2, 62, 76	11, 24, 44	39, 72, 78
24, 35, 78	18, 42, 84	10, 51, 60	25, 35, 46	4, 22, 47
41, 46, 76	39, 73, 79	55, 71, 74	19, 71, 80	3, 55, 67
67, 78, 81	21, 47, 69	20, 24, 66	0, 46, 66	8, 16, 71
6, 34, 57	1, 8, 28	52, 59, 71	22, 51, 73	38, 52, 75
1, 36, 48	8, 12, 56	0, 59, 78	24, 32, 38	8, 32, 53
33, 56, 81	46, 62, 83	9, 54, 62	28, 36, 73	0, 35, 53
32, 72, 77	31, 80, 84	18, 61, 63	56, 59, 70	12, 44, 49
29, 39, 71	14, 25, 38	27, 53, 68	1, 46, 59	7, 53, 71
28, 54, 73	48, 55, 64	49, 52, 81	19, 65, 80	50, 66, 71
46, 69, 82	18, 45, 49	17, 54, 69	16, 70, 76	10, 14, 64
7, 12, 54	24, 65, 72	22, 50, 59	37, 60, 79	45, 54, 57
69, 72, 78	0, 72, 75	7, 9, 27	60, 64, 72	16, 42, 68
16, 46, 79	2, 61, 81	6, 39, 66	3, 28, 39	13, 16, 59
26, 33, 55	36, 38, 68	11, 13, 77	4, 37, 56	12, 60, 76
4, 28, 67	8, 21, 84	16, 26, 51	23, 61, 68	43, 69, 78
20, 35, 43	0, 14, 71	23, 60, 64	14, 42, 44	0, 27, 72
29, 50, 58	20, 55, 60	1, 11, 78	41, 49, 83	2, 10, 60
14, 30, 59	3, 24, 56	26, 53, 61	30, 36, 82	44, 55, 76
18, 39, 59	28, 37, 67	23, 51, 53	24, 35, 61	5, 49, 82

3, 37, 66	30, 66, 73	21, 51, 55	32, 54, 63	7, 10, 54
11, 17, 23	34, 38, 80	49, 57, 78	24, 45, 80	50, 68, 70
19, 41, 77	11, 41, 80	17, 27, 42	49, 62, 76	43, 61, 84
4, 10, 59	0, 9, 46	3, 43, 73	44, 50, 78	42, 70, 73
18, 37, 71	28, 64, 80	19, 39, 61	39, 57, 76	8, 24, 79
1, 46, 52	24, 26, 39	1, 14, 83	0, 20, 33	15, 22, 51
53, 68, 83	17, 55, 70	31, 37, 41	4, 53, 60	4, 32, 57
36, 64, 69	35, 72, 78	17, 32, 45	13, 29, 33	6, 65, 75
35, 61, 80	3, 6, 26	10, 12, 45	57, 70, 72	14, 68, 80
1, 33, 66	2, 40, 73	8, 64, 83	1, 35, 84	20, 25, 32
2, 34, 36	18, 51, 57	7, 32, 83	5, 7, 41	29, 37, 82
11, 52, 75	22, 49, 61	0, 31, 34	7, 12, 72	0, 31, 70
40, 55, 59	1, 3, 49	27, 74, 80	17, 39, 60	17, 19, 51
6, 20, 75	8, 18, 24	43, 45, 52	35, 69, 71	48, 52, 74
8, 38, 82	5, 18, 68	12, 52, 82	28, 59, 78	5, 15, 61
17, 36, 70	41, 45, 71	33, 53, 83	8, 55, 70	2, 13, 79
4, 12, 56	43, 49, 73	0, 58, 67	8, 67, 73	51, 75, 81
2, 17, 43	39, 43, 73	17, 19, 69	11, 40, 47	12, 21, 69
23, 44, 66	23, 53, 57	1, 19, 59	34, 49, 71	28, 33, 61
13, 25, 61	27, 40, 66	58, 64, 68	17, 46, 68	5, 36, 79
12, 44, 71	18, 27, 42	28, 47, 81	18, 29, 76	1, 54, 61
1, 14, 54	5, 25, 38	64, 73, 83	11, 36, 41	2, 25, 64
44, 68, 76	13, 43, 45	7, 20, 68	32, 56, 60	39, 49, 76
21, 35, 50	32, 51, 77	5, 13, 64	9, 21, 45	24, 29, 66
4, 16, 55	41, 50, 63	44, 72, 82	43, 56, 74	10, 20, 30
4, 51, 84	0, 25, 73	0, 33, 49	3, 54, 76	12, 31, 38

3, 30, 68	29, 63, 83	19, 23, 34	9, 38, 68	31, 39, 69
33, 38, 46	40, 71, 81	49, 64, 77	1, 13, 84	44, 56, 79
26, 28, 67	12, 17, 73	1, 49, 73	40, 60, 84	40, 55, 78
26, 42, 58	2, 6, 37	22, 36, 63	20, 30, 42	9, 14, 70
11, 36, 82	5, 28, 58	35, 37, 73	9, 16, 53	2, 66, 74
7, 27, 77	1, 41, 57	21, 79, 84	47, 70, 82	9, 13, 78
17, 61, 73	42, 62, 70	11, 24, 47	66, 76, 79	2, 25, 44
17, 23, 72	10, 70, 84	8, 29, 40	32, 76, 83	6, 12, 81
4, 63, 75	22, 75, 78	11, 63, 68	13, 22, 54	2, 20, 46
32, 39, 69	0, 51, 81	5, 34, 46	6, 13, 55	19, 28, 56
18, 31, 67	37, 47, 83	5, 37, 45	58, 66, 70	15, 31, 65
36, 62, 82	23, 31, 65	5, 79, 83	4, 35, 65	13, 16, 66
2, 5, 51	12, 38, 71	17, 35, 72	1, 8, 82	20, 22, 42
60, 67, 70	20, 63, 68	41, 55, 62	10, 22, 48	28, 43, 66
15, 19, 42	27, 50, 54	11, 15, 17	11, 13, 47	38, 46, 67
22, 33, 51	28, 34, 59	1, 15, 49	9, 11, 58	48, 57, 64
21, 35, 57	43, 47, 55	8, 33, 48	5, 38, 69	0, 50, 60
21, 42, 83	8, 64, 80	29, 69, 81	48, 53, 57	1, 42, 56
21, 39, 70	2, 13, 75	17, 48, 57	61, 65, 75	64, 68, 71
8, 23, 46	5, 12, 20	2, 41, 77	24, 59, 63	5, 21, 64
1, 25, 34	28, 57, 62	21, 33, 58	17, 26, 38	5, 70, 78
2, 27, 57	42, 75, 84	19, 42, 58	41, 59, 82	7, 20, 42
37, 45, 84	15, 35, 74	24, 34, 71	39, 60, 74	18, 58, 77
33, 46, 51	13, 59, 68	12, 77, 83	6, 18, 44	31, 44, 54
70, 75, 83	14, 28, 56	34, 60, 62	7, 19, 48	8, 23, 80
37, 76, 83	6, 18, 56	7, 38, 61	19, 62, 68	13, 46, 52

16, 50, 55	12, 25, 52	22, 29, 77	25, 45, 54	2, 4, 58
5, 31, 66	13, 27, 48	12, 25, 32	26, 38, 54	19, 29, 43
10, 70, 84	18, 44, 82	19, 23, 61	24, 39, 51	30, 64, 71
18, 44, 47	3, 23, 59	33, 41, 61	15, 23, 80	11, 35, 71
15, 23, 84	48, 76, 80	27, 33, 72	43, 54, 77	11, 16, 70
0, 4, 55	0, 3, 22	39, 49, 65	26, 36, 42	11, 27, 73
18, 52, 75	15, 34, 43	35, 44, 62	22, 31, 84	12, 20, 60
35, 42, 82	59, 75, 80	15, 30, 50	8, 27, 34	15, 39, 55
6, 24, 56	2, 33, 77	6, 67, 78	17, 50, 79	14, 22, 63
66, 75, 77	36, 47, 77	61, 66, 70	8, 16, 48	6, 8, 49
37, 48, 79	25, 33, 65	9, 13, 55	32, 41, 53	6, 8, 41
0, 6, 21	13, 32, 74	25, 39, 80	7, 18, 37	16, 31, 82
16, 26, 63	22, 36, 62	32, 55, 80	9, 51, 75	1, 66, 71
8, 36, 61	7, 43, 52	3, 53, 63	4, 11, 14	28, 36, 55
26, 41, 76	6, 22, 56	26, 70, 77	38, 42, 66	10, 52, 74
7, 15, 39	10, 24, 74	2, 17, 31	20, 36, 47	5, 7, 52
32, 68, 70	51, 61, 74	40, 74, 78	2, 28, 49	15, 54, 67
7, 50, 57	4, 18, 58	5, 29, 40	4, 37, 83	3, 7, 48
7, 52, 62	42, 72, 75	6, 77, 79	3, 5, 78	13, 47, 75
46, 52, 73	29, 31, 64	14, 66, 68	3, 10, 79	25, 33, 67
47, 51, 72	37, 61, 68	1, 29, 44	20, 27, 84	50, 57, 80
54, 57, 66	0, 27, 52	42, 49, 80	34, 69, 78	41, 65, 69
1, 57, 65	27, 61, 76	30, 69, 73	26, 69, 79	5, 13, 25
16, 25, 49	40, 52, 78	20, 52, 71	21, 68, 79	8, 42, 57
15, 33, 59	56, 69, 77	2, 6, 52	5, 51, 60	1, 9, 22
20, 59, 67	21, 23, 31	5, 53, 78	18, 26, 34	12, 32, 35

0, 23, 64	17, 67, 78	9, 56, 80	0, 53, 59	9, 38, 76
40, 52, 72	44, 74, 79	10, 31, 59	18, 34, 39	17, 61, 67
34, 59, 81	43, 50, 62	19, 30, 82	17, 37, 48	16, 35, 69
21, 41, 44	54, 78, 80	0, 4, 30	13, 72, 83	10, 47, 80
54, 58, 69	25, 31, 47	39, 73, 77	19, 32, 78	37, 51, 59
23, 56, 58	15, 53, 60	15, 58, 78	15, 36, 54	4, 38, 62
5, 47, 76	14, 38, 78	66, 70, 80	16, 37, 72	43, 70, 75
36, 48, 70	53, 69, 79	10, 22, 54	20, 25, 47	2, 71, 79
5, 32, 46	10, 28, 38	8, 12, 78	45, 60, 66	1, 42, 53
37, 40, 48	1, 10, 31	27, 71, 79	6, 11, 15	8, 53, 58
9, 17, 54	4, 32, 63	8, 20, 59	1, 68, 79	1, 14, 27
19, 27, 63	42, 45, 83	41, 55, 60	23, 38, 78	6, 35, 41
10, 64, 76	33, 57, 83	18, 21, 72	10, 60, 79	12, 14, 46
12, 16, 38	33, 54, 72	19, 66, 75	19, 30, 56	37, 47, 58
7, 44, 71	13, 44, 57	22, 32, 75	0, 3, 56	7, 14, 84
46, 48, 80	57, 62, 70	25, 47, 71	14, 16, 80	2, 28, 82
12, 36, 51	25, 34, 56	3, 16, 36	23, 39, 65	2, 14, 64
15, 35, 58	38, 53, 76	11, 43, 63	4, 10, 81	45, 76, 82
56, 61, 66	18, 62, 84	10, 40, 69	9, 47, 65	39, 62, 80
15, 56, 74	3, 6, 40	14, 20, 32	47, 59, 71	8, 43, 47
36, 41, 67	2, 39, 54	30, 52, 71	47, 55, 81	12, 22, 36
5, 33, 35	26, 41, 46	26, 76, 83	30, 60, 66	14, 30, 34
6, 11, 60	12, 52, 60	7, 43, 58	38, 56, 66	10, 29, 67
12, 66, 84	34, 59, 82	3, 12, 21	35, 68, 79	29, 64, 74
16, 36, 64	6, 20, 70	3, 9, 25	26, 59, 66	9, 33, 64
42, 55, 83	25, 55, 67	2, 24, 64	9, 37, 39	27, 54, 65

9, 57, 84	10, 28, 66	14, 52, 62	22, 41, 60	75, 78, 80
28, 44, 70	38, 51, 76	4, 11, 54	35, 48, 68	40, 77, 80
4, 31, 58	10, 73, 80	28, 48, 78	45, 67, 72	4, 50, 57
26, 37, 68	38, 41, 47	43, 51, 57	28, 60, 75	26, 32, 69
60, 63, 73	29, 70, 79	16, 72, 77	16, 52, 79	19, 32, 45
20, 33, 39	16, 23, 76	28, 31, 34	14, 17, 53	9, 18, 28
32, 73, 82	8, 72, 83	30, 46, 48	0, 40, 66	20, 24, 74
36, 65, 73	17, 20, 41	22, 28, 37	5, 35, 74	32, 43, 76
1, 16, 38	20, 67, 71	22, 73, 84	5, 69, 75	28, 36, 83
9, 42, 70	28, 60, 76	55, 70, 74	5, 42, 49	23, 47, 78
25, 28, 69	16, 32, 80	2, 6, 33	0, 41, 67	4, 25, 31
22, 26, 70	14, 32, 64	29, 78, 83	5, 33, 34	7, 53, 59
30, 48, 56	21, 36, 64	14, 49, 56	14, 57, 78	15, 46, 67
11, 39, 67	5, 16, 26	25, 59, 62	5, 11, 27	45, 48, 72
22, 44, 59	8, 56, 63	28, 46, 64	42, 61, 80	11, 64, 75
21, 30, 41	12, 79, 81	18, 34, 83	35, 62, 64	30, 53, 66
65, 78, 80	19, 29, 73	31, 63, 83	22, 34, 65	1, 70, 74
19, 31, 52	15, 28, 60	11, 25, 83	7, 28, 77	23, 42, 71
35, 39, 41	1, 5, 26	14, 26, 80	38, 42, 52	40, 48, 56
43, 63, 66	23, 52, 81	15, 22, 35	57, 62, 82	39, 47, 79
14, 29, 67	39, 65, 83	10, 43, 79	2, 14, 40	2, 20, 53
51, 57, 59	53, 60, 72	12, 26, 46	14, 39, 72	24, 50, 69
19, 44, 59	3, 12, 42	39, 62, 67	37, 42, 65	2, 21, 44
19, 72, 81	32, 47, 72	20, 36, 81	19, 38, 83	37, 61, 64
11, 16, 45	39, 68, 81	4, 9, 15	25, 62, 74	38, 48, 59
47, 56, 68	12, 34, 39	10, 69, 75	10, 18, 53	7, 51, 59

25, 68, 78	53, 81, 84	7, 60, 82	1, 65, 67	6, 27, 81
26, 28, 37	10, 43, 55	30, 57, 60	51, 64, 84	1, 19, 64
12, 22, 55	25, 31, 79	27, 65, 68	6, 46, 54	67, 71, 83
11, 76, 78	0, 47, 80	0, 29, 79	35, 46, 78	43, 65, 74
11, 20, 80	4, 36, 55	21, 46, 83	27, 43, 69	26, 29, 37
49, 51, 58	10, 25, 77	31, 62, 65	17, 37, 55	12, 41, 47
24, 40, 69	50, 56, 61	29, 73, 75	1, 21, 58	24, 58, 62
50, 73, 82	18, 22, 82	20, 34, 74	27, 31, 70	3, 17, 59
9, 16, 58	31, 33, 60	12, 17, 27	57, 65, 77	11, 24, 75
6, 20, 50	23, 42, 75	20, 49, 74	42, 64, 79	11, 14, 19
18, 33, 38	17, 49, 79	36, 62, 71	9, 69, 73	23, 36, 72
6, 10, 41	13, 36, 84	14, 34, 42	9, 17, 58	32, 36, 38
15, 24, 31	46, 67, 79	6, 42, 54	31, 37, 51	62, 66, 73
45, 50, 81	4, 47, 66	26, 52, 54	41, 56, 70	10, 13, 29
13, 37, 60	6, 32, 54	21, 40, 51	3, 37, 60	24, 34, 83
8, 14, 58	15, 33, 62	10, 37, 77	41, 45, 69	4, 33, 40
32, 34, 81	3, 17, 28	27, 29, 56	1, 48, 53	22, 28, 38
26, 34, 61	33, 42, 73	32, 45, 48	19, 29, 75	41, 64, 66
65, 73, 78	4, 8, 28	6, 16, 52	7, 33, 41	21, 23, 76
29, 52, 80	76, 78, 82	43, 49, 84	20, 35, 53	16, 30, 42
9, 23, 45	13, 31, 73	11, 42, 67	32, 41, 84	39, 71, 74
19, 40, 53	1, 42, 74	7, 48, 81	30, 39, 61	24, 48, 73
13, 28, 51	15, 27, 70	5, 26, 30	26, 56, 72	2, 13, 82
55, 61, 82	32, 44, 52	11, 39, 45	22, 50, 66	8, 46, 61
3, 46, 53	7, 34, 63	73, 77, 82	13, 39, 52	36, 67, 68
25, 45, 50	34, 55, 57	1, 4, 7	24, 26, 28	33, 36, 74

18, 50, 62	13, 61, 70	36, 40, 42	35, 40, 56	1, 16, 24
16, 20, 22	55, 57, 75	12, 24, 30	56, 63, 67	29, 51, 80
21, 56, 62	15, 43, 51	11, 16, 29	9, 12, 24	7, 11, 58
12, 28, 78	18, 22, 60	7, 10, 49	3, 33, 39	7, 34, 60
8, 45, 57	57, 75, 82	3, 28, 81	4, 64, 76	7, 9, 79
24, 36, 59	18, 30, 68	17, 32, 82	34, 42, 69	11, 32, 69
34, 61, 65	23, 55, 82	23, 33, 55	10, 39, 50	8, 20, 56
9, 31, 36	40, 64, 70	27, 62, 71	72, 80, 82	38, 40, 50
9, 32, 61	23, 36, 83	10, 61, 72	25, 33, 69	16, 18, 75
2, 4, 42	60, 74, 76	13, 50, 79	52, 61, 79	0, 57, 69
10, 46, 53	5, 34, 58	8, 15, 66	33, 42, 78	1, 31, 37
13, 17, 58	25, 60, 68	35, 43, 59	4, 44, 69	52, 61, 81
10, 65, 73	0, 38, 84	4, 33, 84	29, 48, 61	23, 58, 63
11, 40, 64	7, 11, 29	11, 19, 49	9, 25, 60	30, 33, 74
24, 46, 83	23, 27, 77	7, 72, 76	14, 21, 73	29, 35, 79
42, 53, 81	0, 62, 84	4, 13, 68	6, 69, 84	52, 57, 65
49, 74, 82	5, 54, 79	17, 23, 25	45, 51, 58	14, 33, 62
40, 58, 76	37, 39, 44	7, 37, 46	21, 50, 72	18, 32, 51
50, 53, 75	0, 57, 78	10, 24, 40	41, 52, 68	7, 37, 55
20, 29, 72	5, 70, 72	11, 38, 83	47, 55, 65	7, 23, 33
5, 19, 82	8, 16, 21	77, 81, 84	30, 39, 75	21, 39, 68
48, 62, 73	8, 39, 70	10, 17, 25	31, 54, 58	13, 41, 63
25, 30, 76	3, 19, 50	6, 15, 44	19, 30, 53	28, 41, 79
30, 34, 63	1, 3, 40	7, 23, 67	35, 51, 62	1, 27, 55
2, 8, 35	42, 68, 74	39, 64, 82	15, 50, 68	11, 34, 73
23, 48, 69	23, 55, 66	31, 60, 71	18, 43, 50	5, 66, 80

22, 40, 83	9, 17, 55	1, 23, 70	14, 49, 68	17, 62, 80
26, 65, 81	0, 40, 50	33, 64, 74	7, 11, 65	6, 34, 47
31, 46, 72	1, 13, 73	19, 48, 68	8, 39, 59	7, 33, 45
6, 32, 76	48, 66, 69	38, 51, 83	5, 44, 60	15, 17, 64
44, 54, 59	46, 55, 82	34, 39, 68	50, 76, 84	18, 27, 83
10, 20, 38	31, 53, 79	2, 72, 74	43, 68, 71	13, 56, 76
1, 47, 68	32, 35, 48	15, 52, 58	23, 27, 74	36, 40, 43
24, 31, 43	18, 56, 75	37, 43, 53	46, 62, 78	4, 70, 81
25, 58, 84	18, 41, 48	32, 34, 58	0, 18, 20	5, 9, 41
14, 30, 43	59, 74, 82	36, 60, 84	7, 22, 31	13, 20, 40
42, 60, 82	17, 60, 65	19, 33, 84	6, 27, 36	26, 36, 75
13, 33, 56	21, 47, 79	35, 53, 67	12, 27, 59	17, 42, 51
20, 69, 77	42, 51, 62	21, 34, 77	16, 68, 81	1, 23, 43
61, 76, 80	24, 45, 60	26, 47, 53	6, 21, 30	1, 5, 74
3, 46, 57	14, 49, 61	30, 75, 77	11, 53, 81	9, 22, 83
8, 38, 71	0, 15, 77	19, 27, 66	21, 37, 71	2, 62, 71
7, 20, 84	27, 60, 81	16, 39, 78	27, 58, 64	18, 46, 54
29, 42, 76	14, 48, 58	4, 66, 79	7, 68, 76	6, 26, 64
28, 52, 66	43, 53, 82	34, 66, 72	14, 59, 79	12, 68, 81
3, 23, 32	5, 30, 58	29, 33, 38	31, 59, 66	0, 12, 83
8, 52, 78	18, 31, 72	2, 42, 48	4, 27, 36	30, 54, 70
44, 58, 81	18, 22, 64	6, 38, 84	3, 13, 84	13, 24, 53

B.2 A Hill Climbing Algorithm

```
import java.util.Random;
```

```
import java.util.Arrays;

/**
* Hill climbing algorithm to find a specified
* triple system.
*
* @author Katherine Perry
* @version 01-27-2014
*/

public class FindTriplesJune2016 {


 /**
 * Uses a hill climbing algorithm on a given adjacency matrix
 * to find a desired triple system, where each pair of non-
 * contiguous points in a 2-dimensional array are in exactly
 * 3 triples.
 *
 * @param args Command line arguments (not used).
 */

   public static void main(String[] args) {

   // creates variables including adjacency matrix and array to store triples

      int[][] adjacencyMatrix = new int[85][85];
      int[][] triples = new int [3400][3];
      int tripleNum = 0;
```

```
  int total = 0;
  String s = "\\";
  String ss = s+s;


// Builds adjacency matrix: puts 3 edges between noncontiguous pairs of
// points and 0 edges between 2-contiguous pairs and eliminates loops

  for (int row = 0; row < adjacencyMatrix.length; row++)
    for (int col = 0; col < adjacencyMatrix[row].length; col++)
      if (row == col) {
        adjacencyMatrix[row][col] = 0;
      }
      else if (row == (col + 1)) {
        adjacencyMatrix[row][col] = 0;
      }
      else if (row == (col - 1)) {
        adjacencyMatrix[row][col] = 0;
      }
      else {
        adjacencyMatrix[row][col] = 3;
        adjacencyMatrix[row][(row + 17) % 85] = 0;
        adjacencyMatrix[row][(row + 68) % 85] = 0;
      }

  //corrections
  adjacencyMatrix[0][16] = 0;
  adjacencyMatrix[16][0] = 0;
  adjacencyMatrix[16][17] = 3;
```

```
adjacencyMatrix[17][16] = 3;
adjacencyMatrix[17][33] = 0;
adjacencyMatrix[33][34] = 3;
adjacencyMatrix[33][17] = 0;
adjacencyMatrix[34][33] = 3;
adjacencyMatrix[34][50] = 0;
adjacencyMatrix[50][51] = 3;
adjacencyMatrix[50][34] = 0;
adjacencyMatrix[51][50] = 3;
adjacencyMatrix[51][67] = 0;
adjacencyMatrix[67][68] = 3;
adjacencyMatrix[67][51] = 0;
adjacencyMatrix[68][67] = 3;
adjacencyMatrix[68][84] = 0;
adjacencyMatrix[84][68] = 0;

//builds array to store triples. defaults all values to 0.

for (int row = 0; row < triples.length; row++)
  for (int col = 0; col < triples[row].length; col++)
    triples[row][col] = 0;

//Checks to see if every entry in the triples matrix has been
//filled. If it hasn't, program continues hill climbing algorithm. If it
//If it has, program prints out set of triples.

while ((triples[3399][0] == 0) && (triples[3399][1] == 0)) {

  // *** HILL CLIMBING ALGORITHM ****
```

```
Random generator = new Random();
int v1, v2, v3;

v1 = generator.nextInt(85);
v2 = generator.nextInt(85);
v3 = generator.nextInt(85);

boolean replaced = false;

int specialTriple, numToReplace;

//Checks to see if edges between v1 and other two vertices still exist
//If they do, either the triple is added, or one triple is deleted and
//then the triple is added
if ((adjacencyMatrix[v1][v2] > 0) && (adjacencyMatrix[v1][v3] > 0)) {

  //if all necessary edges exist, program adds triple to array of
     triples
  //and decreases each corresponding entry in adjacency matrix by 1.
  if (adjacencyMatrix[v2][v3] > 0) {
    //enters triple in array of triples
    triples[tripleNum][0] = v1;
    triples[tripleNum][1] = v2;
    triples[tripleNum][2] = v3;

    //adjusts adjacency matrix
    adjacencyMatrix[v1][v2]--;
    adjacencyMatrix[v1][v3]--;
```

```
    adjacencyMatrix[v2][v1]--;
    adjacencyMatrix[v2][v3]--;
    adjacencyMatrix[v3][v1]--;
    adjacencyMatrix[v3][v2]--;

    //increases number to enter next triple in
    tripleNum++;

    total++;
}

else {
//searches for first triple containing the edge between v2 and v3
//replaces that triple with v1, v2, v3.
    for (int row = 0; row < tripleNum; row++)
      if ((replaced == false) &&
            ((triples[row][0] == v2 && triples[row][1] == v3) ||
            (triples[row][1] == v2 && triples[row][2] == v3) ||
            (triples[row][0] == v3 && triples[row][1] == v2) ||
            (triples[row][1] == v3 && triples[row][2] == v2) ||
            (triples[row][0] == v2 && triples[row][2] == v3) ||
            (triples[row][2] == v2 && triples[row][0] == v3))) {


          specialTriple = row;

            //finds third number in first triple containing the edge
            //between v2 and v3.
```

```
for (int colu = 0; colu < triples[row].length; colu++)
  if (triples[specialTriple][colu] != v2 &&
     triples[specialTriple][colu] != v3) {
    numToReplace = triples[specialTriple][colu];



    //replaces first triple with triple with edge between v2
       and v3
    triples[specialTriple][0] = v1;
    triples[specialTriple][1] = v2;
    triples[specialTriple][2] = v3;

    //adjusts adjacency matrix
    adjacencyMatrix[v1][v2]--;
    adjacencyMatrix[v1][v3]--;
    adjacencyMatrix[v2][v1]--;
    adjacencyMatrix[v3][v1]--;

    adjacencyMatrix[numToReplace][v2]++;
    adjacencyMatrix[numToReplace][v3]++;
    adjacencyMatrix[v2][numToReplace]++;
    adjacencyMatrix[v3][numToReplace]++;

    total++;

    //ends if statement
    replaced = true;
  }
```

```
            }
        }

    }

}

//orders triples in triple matrix

for (int row = 0; row < triples.length; row++) {
  Arrays.sort(triples[row]);
}

  //print triples matrix

for (int row = 0; row < (triples.length - 4); row += 5) {

    System.out.print(triples[row][0] + ",\t" + triples[row][1]
          + ",\t" + triples[row][2] + "& \t\t"
          + triples[row+1][0] + ",\t" + triples[row+1][1] + ",\t"
          + triples[row+1][2] + "& \t\t"
          + triples[row+2][0] + ",\t" + triples[row+2][1] + ",\t"
          + triples[row+2][2] + "& \t\t"
          + triples[row+3][0] + ",\t" + triples[row+3][1] + ",\t"
          + triples[row+3][2] + "& \t\t"
          + triples[row+4][0] + ",\t" + triples[row+4][1] + ",\t"
          + triples[row+4][2] + ss + "\t" + "\\hline" );
```

```
        System.out.println();
      }
System.out.println("Total Number of Triple Tries = " + total);
   }
}
```