Title of Thesis with all Formula,
Symbols, or Greek Letters Written out in Words

by

Patrick Di Salvo

A Thesis

presented to

The University of Guelph

In partial fulfilment of requirements
for the degree of

Master of Pokemon, Poke master

in

Computer Science

Guelph, Ontario, Canada

ABSTRACT

Title of Thesis with all Formula, Symbols, or Greek Letters Written out in Words

Patrick Di Salvo
University of Guelph, 2099

Advisor:
Dr. Sherlock Holmes

Present your abstract here. This template is just an example of an outline to consider for your thesis. It is a good idea to double-check with the current University's thesis requirements here:

`https://www.uoguelph.ca/graduatestudies/current-students/preparation-your-thesis`

Acknowledgments

Present your acknowledgements here.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Amidakuji is a custom in Japan which allows for a pseudo-random assignment of children to prizes [**?**]. Usually done in Japanese schools, a teacher will draw N vertical lines, hereby known as lines, where N is the number of students in class. At the bottom of each line will be a unique prize. And at the top of each line will be the name of one of the students. The teacher will then draw 0 or more horizontal lines, hereby known as *bars*, connecting two adjacent lines. The more bars there are the more complicated (and fun) the Amidakuji is. No two endpoints of two bars can be touching. Each student then traces their line, and whenever they encounter an end point of a bar along their line, they must cross the bar and continue going down the adjacent line. The student continues tracing down the lines and crossing bars until they get to the end of the ladder lottery. The prize at the bottom of the ladder lottery is their prize [**?**]. See Figure **??** for an example of a ladder lottery.

The exact date at which Amidakuji were invented is unknown. However, the word Amidakuji has an interesting etemology. In Japanese, Amida is the Japanese name for Amitabatha, the supreme Buddha of the Western Paradise. See image —image ref— for a picture af Amithaba. Amithaba is a Buddha from India and there is a cult based around him. The cult of Amida, otherwise known as Amidism, believes that by worshiping Amithaba, they shall enter into the his Western Paradie. Amidism began in India in the fourth century and made its way to China and Korea in the fifth centur, and finally came to Japan in ninth century. It was in Japan, where the game Amidakuji began. It is known as 'Ghost Legs' in China and Ladder Lotteries in English.

The game Amidakuji began in Japan in the Muromachi period, which spanned from 1336 to 1573. During the Muromachi period, the game was played by having players draw their names at the top of the lines, and at the bottom of the lines were pieces of paper that had the amount the players were willing to bet. The pieces of paper were folded in the shape of Amithaba's halo, which is why the game is called Amidakuji. Kuji is the Japanese word for lottery. Hence the name of the game being Amidakuji.

1.1 Thesis Statement

This thesis presents solutions to three problems surrounding ladder lotteries. The problems that this thesis presents solutions to are firstly the counting problem, secondly the minimial height problem and thirdly the problem of creating a Gray Code for generating canonical ladders within different sets of ladder lotteries.

1.2 Overview of Thesis

This thesis is broken down into several sections. Firstly, an introuduction to ladder lotteries, and how they pertain to computer science will be presented. This will be followed by a literature review of ladder lotteries in which discussions of solved problems will be provided, along with the commonalities between ladder lotteries and other mathematical objects. Following the literature review, the methodology used to conduct the research will be discussed. This section will focus primarily on the algorithms used to generate the data for this reasearch. Following the methodology section, the findings of the research will be discussed and analyzed. In this section there will be proofs and formulas for certain propositions made about ladder lotteries. This section will be the bulk of the findings for this thesis. Following the results section, a summary of future work will be provided. In this section, an of the failures and successes of this research will be analyzed. There will also be commentary on

Buddha-Weekly-Amitayus-Amitabha-Buddhism.jpg

Figure 1.1: A picture of Buddha Amithaba

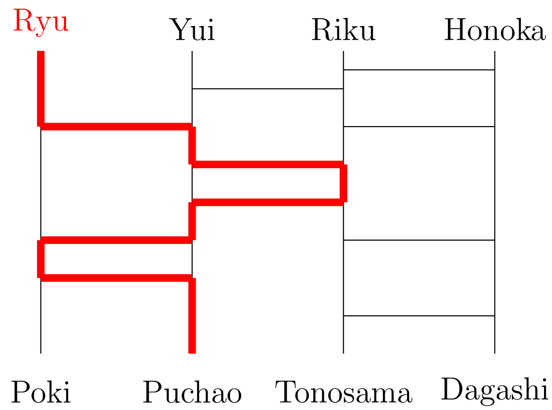

Figure 1.2: A ladder lottery where Ryu gets Puchao, Yui gets Dagashi, Riku gets Tonosama and Honoka gets Poki. You can see that Ryu's path is marked by red bars.

open (unsolved) problems related to ladder lotteries and a discussion of how research on ladder lotteries could be used in other fields. Finally a conlcusion that summarizes the thesis will be provided.

Chapter 2

Background

An interesting property about ladder lotteries is that they can be derived from a *permutation* which is a is a unique ordering of objects. For the purposes on this paper, the objects of a permutation will be integers ranging from $[1 \ldots N]$. *Optimal ladder lotteries* are a special case of ladder lotteries in which there is one bar in the ladder for each *inversion* in the permutation. An *inversion* is a relation between two elements in π, π_i and π_j, such that if $\pi_i > \pi_j$ and $i < j$ then π_i and π_j form an inversion. For example, given $\pi = (4, 3, 5, 1, 2)$, its iversion set is $Inv(\pi) = \{(4, 3), (4, 1), (4, 2), (3, 1), (3, 2), (5, 1), (5, 2)\}$. Every permutation has a unique, finite set of optimal ladder lotteries associated with it. Thus, the set of optimal ladder lotteries associated with π, hereby known as $OptL\{\pi\}$, is the set containing all ladder lotteries with a number of bars equal to the number if inversions in π. See figure **??** for an example of an optimal ladder in $OptL\{(4, 3, 2, 1)\}$. For each optimal ladder in $OptL\{\pi\}$, the N elements in π are listed at the top of a ladder and each element is given its own line. At the bottom of a ladder is the *sorted permutation*, hereby known as the *identity permutation* [**?**]. The identity permutation of size N is defined as follows - $I : (1, 2, 3, \ldots, N)$. Each ladder in $OptL\{\pi\}$ has the minimal number of horizontal bars to sort π into the identity permutation. Each bar in a ladder from $OptL\{\pi\}$ uninverts a single inversion in π exactly once. For the remainder of this paper, only optimal ladder lotteries will be discussed. Therefore when the term ladder lottery is used, assume optimal ladder lottery.

Figure 2.1: Two ladders for the permutation (4, 3, 2, 1). The left ladder is an optimal ladder and the right ladder is not. Therefore the left ladder belongs to $optL\{(4, 3, 2, 1)\}$. The bold bars in the right ladder are redundant, thus the right ladder is not optimal

2.1 Literature Review

2.1.1 Literature Overview

The study of ladder lottieres as mathematical objects began in 2010, in the paper **Efficient Enumeration of Ladder Lotteries and its Application**. The paper was written by four authors, Yamanaka, Horiyama, Uno and Wasa. In this paper the authors present an algorithm for generating all the ladder lotteries of an arbitrary permutation, π. Since this paper emerged, there have been several other paper written directly about ladder lotteries. These papers include **The Ladder Lottery Realization Problem**, **Optimal Reconfiguration of Optimal Ladder Lotteries**, **Efficient Enumeration of all Ladder Lotteries with K Bars**, **Coding Ladder Lotteries** and **Enumeration, Counting, and Random Generation of Ladder Lotteries**.

2.1.2 Efficient Enumeration of Laddder Lotteries and its Application

In their paper, Efficient Enumeration of Ladder Lotteries and its Application, the authors provide an algorithm for generating $OptL\{\pi\}$ for any π, in $\mathcal{O}(1)$ per ladder.

This is the first and only published algroithm for generating $OptL\{\pi\}$. The paper also presents the number of ladder lotteries in $OptL\{(11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1)\}$ which is $5, 449, 192, 389, 984$. This is a very impressive accomplishment for reasons which will be discussed later in the literature review.

The authors' algorithm is based on several key concepts, the most important of which is the *local swap operation*. This is the minimal change operation that transitions from one ladder in $OptL\{\pi\}$ to the next ladder. The local swap operation is essentially a 180 degree rotation of three bars in the ladder, all at different levels in the ladder, such that the bottom bar is rotated to the top, the middle bar stays in the middle and the top bar is rotated to the bottom. If the bars undergo a 180 degree rotation to the right, then this is known as a *right swap operation* and if the bars udergo a 180 degree rotation to the left then this is known as a *left swap operation*. To go to the next ladder in the set, the current ladder, L_i udergoes a right swap operation to get toe ladder L_{i+1}. See figure –ref– for an exmaple of a local swap operation.The *route* of an element is the sequence of bars in the ladder, from top left to bottom right, that an element must cross in order to reach its correct position in the identity permutation. Note that each bar has two elements that cross it, therefore the bar belongs to the route of the greater of the two elements. It is important to note that when a right swap operation occurs, two of the three bars belong to the route of a greter element and one bar belongs to the route of a lesser element. Once rotated, the bar of the lesser element is above the bars of the greater element.

The *clean level* refers to the smallest element in π such that none of its bars have undergone a right swap operation. If there is no such element, then the clean level is the maximum element in $\pi + 1$. The *root ladder* is the only ladder in the set with a clean level of 1; in other words, the root ladder is the only ladder in which no bars have undergone a right swap operation. The root ladder is unique to $OptL\{\pi\}$. To see the root ladder of $OptL\{(4, 5, 6, 3, 1, 2)\}$ please refer to figure **??**. Since none of the bars in the root ladder have undergone a right swap operation, it is the only ladder in $OptL\{\pi\}$ that has a clean level of 1. The root ladder is also the original decendant ladder in the set. Insofar as the enumeration algorithm is based on performing a right swap operation on a pervious ladder, then every other ladder must have at least one right swap operation. Since the root ladder has no right swap operations, then it must be the decendant of every other ladder.

The algorithm for generating $OptL\{\pi\}$ in the paper was the backbone of the research for this thesis. However, the algorithm in this paper had some issues. These issues presented several issues during the research for this thesis. The first issue in the paper is that the authors do not provide an algorithm for generating the root ladder in $OptL\{\pi\}$. Seeing as every other in the set is derived from the root ladder, it is essential to build the root ladder without performing a right swap operation. Yet the paper does not provide such an algorithm. The second issue in this paper is that there is no algorithm for permforming a local swap operation on a ladder. Although the authors do a good job explaining under what conditions a local swap operation can be performed, the actual operation itself is trickier than it seems. The last issue in the paper is that it contains an error. The error is that one of the diagrams is incorrect. The diagram is of all the ladders in $OptL\{(5, 6, 3, 4, 2, 1)\}$. This diagram contains 76 ladders when there are actually only 75. The error was confirmed by the author Yamanaka in an email correspondence he and I had. These issues will be resolved in the Methodology and Implementation section.

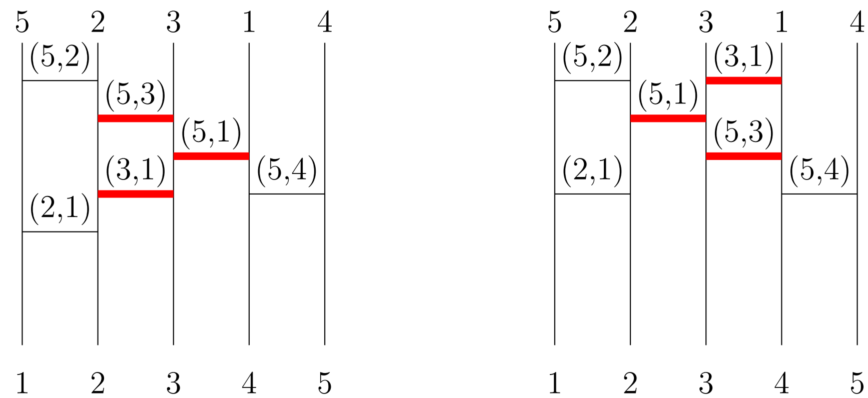

Figure 2.2: Example of a local swap operation. When a right swap operation is permformed on the left ladder, the result is the right ladder. When a left swap operation is permformed on the right ladder, the result is the left ladder.

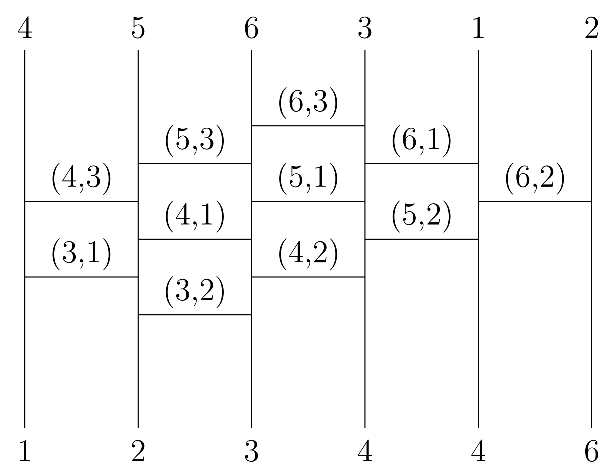

Figure 2.3: The root ladder for $OptL\{(4,5,6,3,1,2)\}$. Notice how none of the bars have undergone a right swap operation. This is clear when considering that there is no bar of a lesser element above the bar(s) of a greater element.

2.1.3 Ladder-Lottery Realization

In their paper **Ladder-Lottery Realization** the authors provide a rather interesting puzzle in regards to ladder lotteries. The puzzle is known as the ladder-lottery realization problem. In order to understand the problem, one must know what a *multi-set* is. A *multi-set* is a set in which an element appears more than once. The exponent above the element indicates the number of times it appears in the set. For example, given the following multi-set, $\{3^2, 2^4, 5^1\}$ the element 3 appears twice in the set, the element 2 appears four times in the set and the element 5 appears once in the set. The ladder-lottery realization puzzle asks, given an arbitrary starting permutation and a multi-set of bars, is there a *non-optimal* ladder lottery for the arbitrary permutation that uses every bar in the multi-set the number of times it appears in the multi-set. For an example of an affirmative solution to the ladder lottery realization problem, see figure –fig ref –.

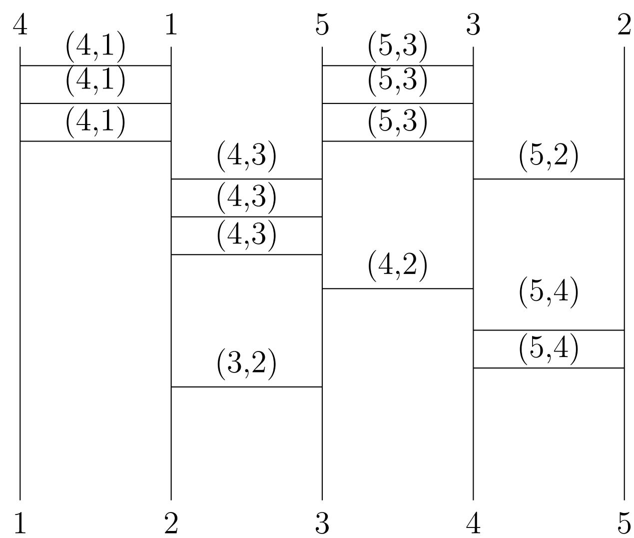

Figure 2.4: An affirmative solution to the Ladder Lottery Realization Problem given a starting perumtation $(4, 1, 5, 3, 2)$ and the multi set of bars $\{(4, 1)^3, (4, 3)^3, (4, 2)^1, (5, 4)^2, (5, 3)^3, (5, 2)^1, (3, 2)^1\}$

The authors prove that the ladder-lottery realization problem in NP-Hard by reducing the ladder-lottery realization to the One-In-Three 3SAT, which has already been proven to be NP-Hard. The One-In-Three 3SAT problem is a problem such that given a set of variables (X), a collection of *disjuntcive clauses* (C) which are disjunctive expressions of over literals of X. Each clause in C contains exactly must contain three literals then is there a truth assignment for X such that each clause in C has exactly one true literal. For eaxmple, let $X = \{p, q, r, s, t\}$ and let $C = \{C_{p,q,s}, C_{r,q,s}C_{p,s,t}, C_{r,t,q}\}$, the question is whether it is possible for each clause to have exactly one true literal. The answer in this case is yes. If $p = T$, $r = T$, $q = F$, $s = F$ and $t = T$ then all the clauses in C have exactly one true literal. The authors reduce the ladder lottery-realization problem to the One-In-Three 3SAT problem by devising four gadgets. The result of the reduction is that the arbitrary starting permutation is equivelent to a derivation of the intial set of variables, X, in the One-In-Three 3SAT problem and the multi-set of bars is equivelent to a derivation of the intial set of clauses, C, in the One-In-Three 3SAT problem.

The authors note that there are two cases in which the ladder-lottery realization problem can be solved in polynomial time. These cases include the follwing. First, if every bar in the multi-set appears exactly once and every bar corresponds to a permutation, then an affirmative solution to the ladder-lottery realization instance can be demonstrated in polynomial time. Second, if there is an inversion in the perumutation and its bar appears in the multi-set an even number of times, then a negative solution to the ladder-lottery realization instance can be solved in polynomial time.

2.2 Optimal Reconfiguration of Optimal Ladder Lotteries

In **Optimal Reconfiguration of Optimal Ladder Lotteries**, the authors provide a polyminomial solution to the minimal reconfiguration problem. The problem states that given two ladder is $OptL\{\pi\}$, L_i and L_m, what is the minimal number of local swap operations to perform that will transition from L_i to L_m. The authors do so based on the local swap operations previously discuessed along with some other concepts. The first of these concepts is termed the *reverse triple*. Basically, a reverse triple is a relation between two arbitrary ladders, L_i, L_m, such that if a bar is right swapped in L_i or L_m and the bar forms a reverse triple between L_i and L_m then the bar must be left swapped in L_m or L_i. Or if it is left swapped in L_i or L_m and the bar forms a reverse triple between L_i and L_m then the bar must be right swapped in L_m or L_i. Essentially, a reverse triple means that if a bar is right or left swapped in one ladder, then it must be the opposite in another ladder for bar to create a reverse triple between the two ladders. The second of the concepts is the *improving triple*. The improving triple is essentially a bar that can be left/right rotated such that the result of the rotation the bar removes a reverse triple between two arbitrary ladders L_i and L_m. The solution tor transition from L_i to L_m with the minimal length reconfiguration sequence is achieved by applying improving triple to the reverse triples between L_i and L_m. That is to say, the length of the reconfiguration sequence is equal to the number of reverse triples between L_i and L_m

The second contribution of this paper is that it provides a closed form upper bound for the minimal length reconfiguration sequence for any permutation of size N. That is to say, given any permutation, π, of size N what is the maximum length of a minimal reconfiguration sequence between two ladders in $OptL\{\pi\}$. The authors prove that it is $OptL\{\pi_{N,N-1,\ldots,1}\}$ that contains the upper bound for the minimal length reconfiguration sequence between two ladders L_i and L_m. Moreover, it is only the root ladder and terminating ladder in $OptL\{\pi_{N,N-1,\ldots,1}\}$ whose minimal reconfiguration sequence is equal to the upper bound. That upper bound is $N\binom{(N-1)}{2}$. This is because the number of reverse triples between the root ladder and the terminating ladder in $OptL\{\pi_{N,N-1,\ldots,1}\}$ is equal to $N\binom{(N-1)}{2}$. Thus, in order to reconfigure the root to the terminating ladder, or vice versa, each reverse triple between them must be improved.

2.3 Efficient Enumeration of all Ladder Lotteries with K Bars

In this paper, the authors apply the same algorithm used in Efficient Enumeration of Optimal Ladder-Lotteries and its Application for generating all ladder lotteries with k bars. The number of elements in The inversion set of π also known as $Inv\{\pi\}$ provides the lower bound for K and the upper bound is positive infinity. Therefore $K = [|Inv\{\pi\}| \ldots N]$

2.4 Coding Latter Lotteries

2.4.1 Overview

In this paper, the authors provide three methods to encode ladder-lotteries as binary strings. Coding discrete objects as binary strings is an appealing theme because it allows for compact represntation of them for a computer.

2.4.2 Route Based Encoding

The first method is termed *route based encoding method* in which each route of an element in the permutation has a binary encoding. Let L_k be a ladder lottery for some arbitrary permutation $\pi = (p_1, \ldots, p_N)$. The route of element p_i is encoded by keeping in mind p_i crosses bars in its route going left zero or more times and crosses bars in its route going right zero or more times. The maximum number of bars p_i can have is $N - 1$, therefore the upper bound for the number of left/right crossings for p_i is $N - 1$. Let a left crossing be denoted with a $'0'$ and let a right crossing be denoted with a $'1'$. Let C_{pi} be the route encoding for the i^{th} element in π. To construct C_{pi}, append 0 and 1 to each other representing the left and right crossings of p_i from the top left to bottom right of the ladder. If the number of crossings for p_i is less than $N-1$, append $0s$ to the encoding of the route of p_i until the encoding is of length $N-1$. Let LC_L be the route encoding for some arbitrary ladder in $OptL\{\pi\}$ is $C_{p_1}, C_{p_2, \ldots C_{p_N}}$. For an example of the route encoding for the root ladder of $(3, 2, 5, 4, 1)$ refer to Fig. 2.5. In Fig 2.5 you will see that C_{p_1} is $11\underline{00}$. Underlined $0s$ are the $0s$ added to ensure the length of C_{p_1} is $N-1$. Since the length of C_{pi} is $N-1$ and the number of elements in π is N then the length of $LC_L = N(N-1)$. Hence the number of bits needed for LC_L belongs to $\mathcal{O}(n^2)$.

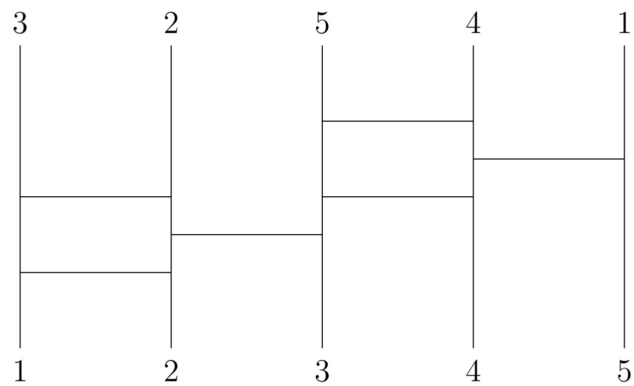

Figure 2.5: The route encoding for the following ladder lottery is $11\underline{00}01\underline{00}11\underline{00}01\underline{00}0000$

2.4.3 Line Based Encoding

The second method is termed *line based encoding* which focuses on encoding the lines of the ladder-lottery. Each line is represented as a sequence of endpoints of bars. Let L be an optimal ladder-lottery with N lines and B bars, then for some arbitrary line i there are zero or more right/left endpoints of bars that come into contact with line i. Let 1 denote a left end point that comes into contact with line i and let 0 denote a right end point that comes into contact with line i. Finally, append a 0 to line i to denote the end of the line. Then line i can be encoded, from top to bottom, as a sequence of $1s$ and $0s$ that terminates in a 0. Let lc_i be the line encoding for line i. Given the ladder in Fig. 2.5, lc_3 is $001\underline{0}$. The $\underline{0}$ denotes the end of the line. Let LC_L be the line encoding for some arbitrary ladder, then $LC_L = lc_1, lc_2, \ldots lc_n$. Let $L_{2.5}$ refer to the ladder in Fig. 2.5, then $LC_{L_{2.5}} = 11\underline{0}010\underline{0}110\underline{0}010\underline{0}0\underline{0}$

In order to reconstruct L_k from LC_{L_k}, or in other words decode LC_{L_k} it is important to recognize that the first line only has left endpoints attached to it. Since left end points are encoded as a 1 then it is guarenteed that the first 0 represents the end of line 1. It is also important to note that for any line $i+1$, if line $i+1$ has a 0 then there must be a corresponding 1 in line i. That is to say, if the right end point of a bar is on line $i+1$ then that same bar must have a left endpoint on line i. To decode LC_L start by decoding line 1. The line will contain 0 or more left end points. To decode lc_{i+1} where $i+1 > 1$, go to lc_i and match each 1 in lc_i with a 0 in lc_{i+1}. Let $k =$ the number of $1s$ in lc_i. Let $j =$ the number of $0s$ in lc_{i+1} then $k = j - 1$; due to the last 0 in lc_{i+1} denoting the end of line $i+1$. Intuitively, this means match every left end point of a bar in line i with a right end point in line $i+1$. The last 0 represents the end of line $i+1$. For the $1s$ in lc_{i+1} draw a left end point on line $i+1$ relative to where the 1 occured to its left and right neigbor in lc_{i+1}. For an example of a full decoding of $LC_{L_{(4,2,3,1)}}$ please refer to Fig. 2.6.

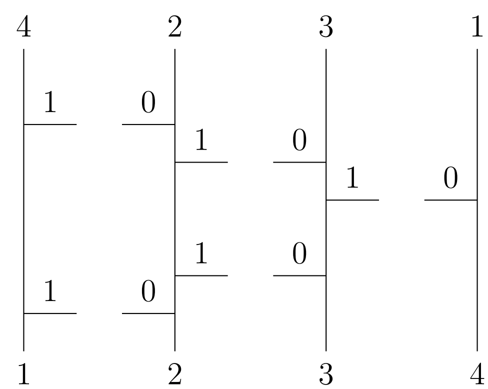

Figure 2.6: $LC_{L_{(4,2,3,1)}} = lc_1 = 11\underline{0}, lc_2 = 0110\underline{0}, lc_3 = 010\underline{0}, lc_4 = 0\underline{0}$

2.4.4 Ladders and Adjacent Transpositions

A ladder lottery is a way of sorting a permutation, yet it can also be thought of as a decomposition of a permutation into *adjacent transpositions*. [**?**] An *adjacent transposition* is simply a swap of two adjacent elements in a permutation. For example, given the permutation (1, 3, 4, 2), an adjacent transposition could be done on the following pairs of elements: (1, 3), (3, 4) or (4, 2). Each would result in a unique permutation. Simply put, given any arbitrary starting permutation, π, keep swapping adjacent inversions until the identity permutation is reached. An optimal ladder lottery from $\pi's$ optimal ladder set is a minimal sequence of adjacent transpositions such that π is sorted into the identity permutation; each ladder in the set represents a sequence of adjacent transpositions for sorting π into the identity permutation. For example, given the permutation (4, 3, 2, 1) there exists eight ladders in this permutation's optimal ladder set. Two of these ladders are found in **??**:

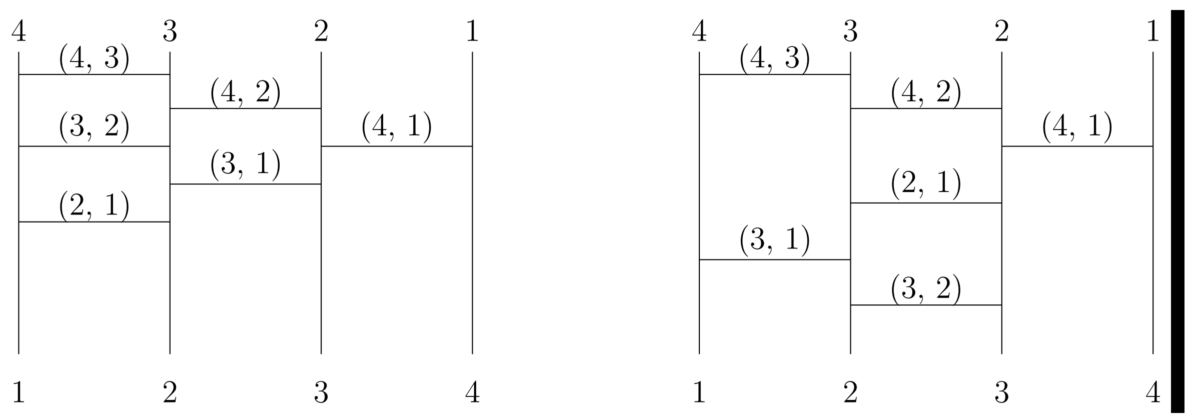

Figure 2.7: The left ladder is one of eight unique ladders from (4,3,2,1)'s optimal ladder set. The right ladder is another one of eight unique ladders form (4,3,2,1)'s optimal ladder set

From looking at the above ladders, going from top left to bottom right, the left ladder represents the sequence of adjacent transpositions (4,3), (4,2), (4,1),(3,2),(3,1),(2,1) whereas the right ladder represent the sequence of adjacent transpositions (4, 3),(4, 2),(4, 1),(2, 1),(3, 1),(3, 2). Notice how the length of the sequences are the same,because both lengths are equal to the minimal number of swaps to sort $(4, 3, 2, 1)$ it is simply the order in which the adjacent transpositions occur in the sequence that makes the sequences different from each other.

Chapter 3

Methodology and Implementation

Chapter 4

Evaluation

Chapter 5

Summary and Future Work

Conclude your thesis with a re-cap of your major results and contributions. Then outline directions for further research and remaining open problems.