Permutations with forbidden subsequences; and, Stack-sortable permutations

by

JULIAN WEST

B.Sc., California Institute of Technology (1985)

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1990

Signature of Author ....
Department of Mathematics
3 July 1990

Certified by ....
Richard P. Stanley
Professor of Applied Mathematics
Thesis Supervisor

Certified by ....
Daniel J. Kleitman
Chair, Applied Mathematics Committee

Accepted by ....
Sigurdur Helgason
Chair, Departmental Graduate Committee

Permutations with forbidden subsequences; and, Stack-sortable permutations

JULIAN WEST

Submitted to the Department of Mathematics on 3 July 1990,
in partial fulfillment of the requirements for the Ph.D.

In the first half of this thesis, we consider the problem of counting the number of permutations of length n which have no subsequence having all the same pairwise comparisons as a given excluded subsequence of length k. For $k = 3$, the difficult case is to explain why the pemutations with no pattern of type 123 are equinumerous with those with no pattern of type 132. We present three direct bijections explaining this curious fact. The first of these bijections is due to Rodica Simion and Frank Schmidt; the second to Dana Richards; the third, making use of rooted trees, is introduced here. One extension of the rooted trees approach provides an asymptotic enumeration of a class of the vexillary permutations. We suggest some other possible extensions of this technique. We also generalize the Simion-Schmdit bijection from the forbidden permutation 123 to any forbidden permutation ending with two fixed elements. We also conjecture that there is a further extension to any permutation ending with r fixed elements.

In the second half of the thesis, we examine the related question of sorting permutations on a stack. We seek to generalize from the well-understood problem of sorting on one stack to a problem of sorting on several stacks. We introduce a "sorting function" which sorts all those permutations which can be sorted on one stack. We use the sorting function to define the sorting tree $T(n)$, whose vertices are the permutations of length n, and whose covering relations are defined by the sorting function. A permutation appears on the kth level of $T(n)$ if it requires k applications of the stack-sorting function to be completely sorted. We examine the structure of the sorting tree, giving conditions which locate a permutation above, or below, a certain level. We determine the number of permutations which are children of a given member of one of three classes of permutations. We define the sorting sequence of a permutation to be the sequence of operations which are performed in the application of the sorting algorithm; we then classify the permutations which require a given sorting sequence. For certain types of sorting sequence, we are able to locate all of the associated permutations by level in the sorting tree. A main outstanding conjecture regards the number of permutations which are sortable by two applications of the sorting algorithm; we suggest several possible approaches to proving this conjecture.

Thesis supervisor: Dr. Richard P. Stanley, Professor of Applied Mathematics

Acknowledgments

I should like to thank everyone, beginning of course with my parents, who ever taught me any of the wonder of knowledge and the joy of mathematics. It does not seem practicable to attempt to recall all of these people by name, but I might just mention Leila Nair, a boyhood tutor; Tony Parker-Jervis, a teaching legend of St. George's School; Ed Barbeau, coach of the Canadian Olympiad team; Herb Ryser, who taught my first full course in combinatorics and was a strong influence in my decisions about a graduate career; T.Sós Vera, who taught an exemplary course in Combinatorics during my term in Budapest.

My advisor, Richard Stanley, first suggested the problems addressed here, and drew my attention to the paper in which Simion and Schmidt effectively demolish the case of forbidden subsequences of length three. He was responsible for encouraging me in all those enquiries which have proved fruitful, and equally for redirecting me at times when I was barking up the wrong tree.

The other members of my committee, Ira Gessel and Daniel Kleitman, also deserve thanks for their guidance. Kleitman's seminars also formed a significant part of my graduate preparation.

Ken Bogart was of significant assistance during my year at Dartmouth College. Several of the results in this thesis were partially developed during conversations in his office. Other ideas, many of which found their way into this work, were developed during discussions with Lauren Rose, Neil Calkin, Jim Driscoll, Dave Wagner and Robin Pemantle.

In addition, the following mathematicians have either looked over work in progress or suggested future problems: Anders Björner, Persi Diaconis, Erdős Pal, Ira Gessel, Rodica Simion, Tom Trotter.

Contents

1 Introduction to forbidden subsequences **7**
- 1.1 Basic definitions and overview . . . 7
- 1.2 The three standard bijections . . . 9

2 Correspondences between $S_n(123)$ and $S_n(132)$ **12**
- 2.1 Enumeration of $S_n(231)$ . . . 12
- 2.2 Enumeration of $S_n(123)$ . . . 15
- 2.3 The Simion-Schmidt bijection . . . 18
- 2.4 The Richards bijection . . . 20
- 2.5 A Catalan tree . . . 24
  - 2.5.1 In connection with forbidden subsequences . . . 25
  - 2.5.2 In relation to some minimal semiorders . . . 28

3 What is known for $k > 3$ **31**
- 3.1 Generalizations of the Catalan trees . . . 31
  - 3.1.1 Three trees for $k = 4$ . . . 31
  - 3.1.2 Two trees for each $k > 4$ . . . 39
- 3.2 A generalization of the Simion-Schmidt algorithm . . . 40
- 3.3 Tables for $k = 4, 5, 6$ . . . 44
- 3.4 Future directions . . . 49
- 3.5 A recurrence for $k = 4$ . . . 50

4 Introduction to stack-sortable permutations **62**
- 4.1 The sorting algorithm . . . 62
- 4.2 The sorting tree . . . 67
  - 4.2.1 Some basic lemmata . . . 68
  - 4.2.2 Tree depth and the bottom levelsum . . . 70
  - 4.2.3 A necessary, and a sufficient, condition . . . 71
  - 4.2.4 A conjecture about level 2 . . . 76
- 4.3 Unasked questions . . . 78

5 More about the sorting tree **82**
- 5.1 The average depth . . . 82
- 5.2 Results on fertility . . . 85
- 5.3 A subtree isomorphism . . . 92

5.4 Two questions about bracketing sequences 95
5.5 The spectrum of a bracketing sequence 97
5.6 What the spectra say about level 2 105
5.7 Other attempts to prove the level 2 conjecture 108
5.7.1 Appeal to forbidden subsequences 108
5.7.2 A set of recurrence equations 108
5.7.3 Generalized bracketing sequences 109
5.7.4 A paper of Tutte's 110

1 Introduction to forbidden subsequences

1.1 Basic definitions and overview

We regard a permutation $\pi \in S_n$ as a sequence of n elements: $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$. We say that π contains the 3-letter pattern 231 iff there is a triple $1 \leq i < j < k \leq n$ such that $\pi(k) < \pi(i) < \pi(j)$. Otherwise π *avoids* the pattern.

We define τ-avoiding permutations similarly for every $\tau \in S_k$.

Definitions 1.1.1 *For* $\tau = (\tau(1), \tau(2), \ldots, \tau(k)) \in S_k$, *a permutation* $\pi = (\pi(1), \pi(2), \ldots, \pi(n)) \in S_n$ *is* τ*-avoiding iff there is no* $1 \leq i_{\tau(1)} < i_{\tau(2)} < \cdots < i_{\tau(k)} \leq n$ *such that* $\pi(i_1) < \pi(i_2) < \cdots < \pi(i_k)$.

Such a $(\pi(i_{\tau(1)}), \pi(i_{\tau(2)}), \ldots, \pi(i_{\tau(k)}))$ *is called a subsequence of (standard) type* τ.

We can extend these definitions in a straightforward way to any sequence of n distinct positive integers, $\pi = (\pi(1), \pi(2), ..., \pi(n))$. Since π is not, strictly speaking, a permutation unless the set $\{\pi(1), \pi(2), ..., \pi(n)\} = \{1, 2, ..., n\}$, we will refer to such a general sequence as a *permutation sequence* or simply a *sequence*.

We may also say, somewhat abusively, that a subsequence ρ is of type τ if ρ and τ are of the same standard type but τ is a permutation sequence of some set of integers other than $[n]$.

Since we will almost always wish to refer to standard types, we will usually omit the qualifier and refer to these simply as types. This should not lead to confusion.

Example 1.1.2 *In the permutation* $\pi = (4, 5, 2, 3, 1)$, *the subsequence* $(4, 2, 3)$ *could be said to be of type* $(4, 2, 3)$, *but is of standard type* $(3, 1, 2)$. *We will write this as* 312 *whenever it is convenient to do so, and say that* π *contains a subsequence of type* 312. *In either case, the subsequence* $(\pi(2), \pi(3), \pi(4)) = (5, 2, 3)$ *is of the same type.*

The permutation π *also contains four descending subsequences of length three; these are of type* 321. π *also contains four subsequences of type* 231; *subsequences of this type are often referred to as* wedges. *There are no subsequences of types* 123, 132, *or* 213; π *is said to avoid these patterns.*

Two permutation sequences, π, ρ of length n are evidently of the same type iff they have the same pairwise comparisons throughout, namely if $\pi(i) < \pi(j) \iff \rho(i) < \rho(j)$, for all $1 \leq i, j \leq n$.

We denote by $S_n(\tau)$ the set of all permutations in S_n which avoid τ. Also let $S(\tau) = \cup_{n=1}^{\infty} S_n(\tau)$. This is also somewhat abusive notation, as in defining $S(\tau)$ we are abandoning any pretense of interest in the group-theoretical aspects of permutations, yet the letter S is retained because it stands for symmetric group.

Example 1.1.3 *The permutation of the above example, $\pi = (4,5,2,3,1)$, does not belong to the set $S_n(321)$ because it contains descending subsequences of length three. On the other hand, π avoids 213, so $\pi \in S_n(213)$. Also, $\pi \in S_n(123) \cap S_n(132)$.*

Fundamental questions are to determine $|S_n(\tau)|$ viewed as a function of n, and if $|S_n(\sigma)| = |S_n(\tau)|$ for $\sigma \neq \tau$ to discover an explicit bijection between $S_n(\sigma)$ and $S_n(\tau)$. In the next section, we shall see some straightforward examples of such bijections. In chapter two, we consider the case of forbidden subsequences of length 3. In particular, we present three direct bijections between $S_n(123)$ and $S_n(132)$. Two of these are due to Rodica Simion and Frank Schmidt [18] and to Dana Richards [14]; the third, making use of rooted trees, is introduced here.

In chapter three we seek to extend these bijections to forbidden subsequences of arbitrary length. We develop some extensions of the rooted trees approach, and conjecture the existence of some others. This technique provides a bijection between the permutations with no ascending subsequence of length 4 and the *vexillary permutations* of $S_n(2143)$. A second conjecture suggests the possibility of an *injection* based on the rooted trees approach. We generalize the Simion-Schmdit bijection from the forbidden permutation 123 to any forbidden permutation ending with two fixed elements. (This provides a second correspondence between $S_n(1234)$ and the vexillary permutations of $S_n(2143)$.) We also conjecture that there is a further extension to any permutation ending with r fixed elements. In section 3.5, we examine a recurrence relation for $S_n(1234)$.

The subject of restricted permutations is closely linked to the question of sorting permutations on a stack. In fact, we will see in section 2.2 that $S_n(231)$ is the set of *stack-sortable* permutations. We will return to the topic of sorting on stacks in the second half of this thesis. We wish to generalize from the well-understood problem of sorting on one stack to a problem of sorting on several stacks. We choose to do this in the most restrictive way possible. In chapter four, we introduce the sorting function, and define the sorting sequence associated to a permutation. We use the sorting function to define the sorting tree $T(n)$, whose vertices are the permutations of length n, and whose covering relations are defined by the sorting function. A permutation appears on the kth level of $T(n)$ if it requires k applications of the stack-sorting function to be completely sorted. We give conditions which locate a permutation above, or below, a certain level in the sorting tree. The final section of chapter four details alternative generalizations of the problem of sorting on a stack.

Chapter five is devoted to the study of the sorting tree. We determine the number of permutations whose sorted form is a given member of one of three classes of permutations. We then classify the permutations which require a given sorting sequence. For certain types of sorting sequence, we are able to locate all of the associated permutations by level in the sorting tree. A main outstanding conjecture regards the number of permutations which appear on levels 0, 1 and 2 of the sorting tree. In the final section of chapter five we suggest several possible approaches to proving this conjecture.

1.2 The three standard bijections

We present first three basic lemmata which produce numerous bijections between sets of the types $S_n(\sigma)$ and $S_n(\tau)$, $\sigma, \tau \in S_k$ for general k. (Lemma 1 in [18].)

We require the following definitions:

Definitions 1.2.1 *For any $\pi \in S_n$, its reversal $\pi^{|} \in S_n$ is given by $\pi^{|}(i) = \pi(n+1-i)$; its complement $\pi^{-} \in S_n$ is given by $\pi^{-}(i) = n+1-\pi(i)$.*

We use π^{-1} to denote the usual group-theoretic inverse, i.e. $\pi^{-1}(j) = i$ if $\pi(i) = j$.

Notice that the three operations of reversal, complementation and inversion are involutions; that is, $(\pi^{|})^{|} = \pi$, $(\pi^{-})^{-} = \pi$, and $(\pi^{-1})^{-1} = \pi$.

For fixed n, the operation of reversing a permutation sequence π of length n amounts to applying a fixed permutation, depending only on n, to the elements of π. Clearly, the application of one and the same permutation to two sequences of the same type will yield two sequences of the same type. As a special case of this observation, two permutation sequences σ and τ are of the same type if and only if $\sigma^{|}$ and $\tau^{|}$ are of the same type.

For π of length n, $\pi^{-}(i) = (n+1) - \pi(i)$, and $\pi^{-}(j) = (n+1) - \pi(j)$. It is clear, therefore, that $\pi^{-}(i) < \pi^{-}(j)$ if and only if $\pi^{-}(i) > \pi^{-}(j)$. That is, the operation of complementation reverses all the pairwise comparisons between elements of a sequence. Since two sequences are of the same type exactly when they have the same pairwise comparisons, it is clear that σ and τ are of the same type iff σ^{-} and τ^{-} are of the same type.

The observations of the two preceding paragraphs are trivial, but essential to the following lemmata, which can be considered as generalizations of the above. The lemmata, in turn, are straightforward but important aspects of the theory of forbidden subsequences.

Lemma 1.2.2 *If* $\pi \in S_n(\tau)$, *then* $\pi^{\mathsf{l}} \in S_n(\tau^{\mathsf{l}})$.

Proof: Consider an arbitrary subsequence of $\pi \in S_n$, such as $\sigma = (\pi(i_1), \pi(i_2), ..., \pi(i_k))$, where $1 \leq i_1 < i_2 < ... < i_k \leq n$. If σ is of type τ, then $\sigma^{\mathsf{l}} = (\pi(i_k), \pi(i_{k-1}), ..., \pi(i_1))$ is of type τ^{l}. But then σ^{l} is a subsequence of π^{l}, namely $(\pi^{\mathsf{l}}((n+1)-i_k), \pi^{\mathsf{l}}((n+1)-i_{k-1}), \dots, \pi^{\mathsf{l}}((n+1)-i_1))$. We have observed that π^{l} contains the subsequence τ^{l} if π contains the subsequence τ. Since the operation of reversal is an involution, the argument works equally in the opposite direction, and we can replace 'if' by 'if and only if' in the previous sentence. Finally, since a permutation avoids a pattern τ exactly when it does not contain a subsequence of type τ, the proof is complete. □

Lemma 1.2.3 *If* $\pi \in S_n(\tau)$, *then* $\pi^- \in S_n(\tau^-)$.

Proof: If, for $1 \leq i_1 < i_2 < ... < i_k \leq n$, a subsequence $\sigma = (\pi(i_1), \pi(i_2), ..., \pi(i_k))$ is of type τ, then the subsequence $(\pi^-(i_1), \pi^-(i_2), ..., \pi^-(i_k))$ has all its pairwise comparisons the reverse of those of σ. That is, it is of type τ^-. The remainder of the argument is identical in its essentials to the previous proof. □

Lemma 1.2.4 *If* $\pi \in S_n(\tau)$, *then* $\pi^{-1} \in S_n(\tau^{-1})$.

Proof: Suppose π has a subsequence of type τ, namely $(\pi(i_{\tau(1)}), \pi(i_{\tau(2)}, \dots, \pi(i_{\tau(k)}))$, where $1 \leq i_{\tau(1)} < i_{\tau(2)} < \cdots < i_{\tau(k)} \leq n$ and $\pi(i_1) < \pi(i_2) < \cdots < \pi(i_k)$. In light of the last set of inequalities, it is clear that one subsequence of π^{-1} is $(\pi^{-1}(\pi(i_1)), \pi^{-1}(\pi(i_2)), \dots, \pi^{-1}(\pi(i_k))) = (i_1, i_2, ...i_k)$. This is a subsequence of type τ^{-1}. Since π^{-1} contains a sequence of type τ^{-1} precisely when π contains the pattern τ, the inverse permutation π^{-1} avoids τ^{-1} precisely when π avoids τ. □

As a further observation of the relation between the three operations of reversal, complementation and inversion, note that $(\pi^-)^{-1} = (\pi^{-1})^{\mathsf{l}}$.

Example 1.2.5 *In the permutation* $\pi = (\pi(1), \pi(2), \pi(3), \pi(4), \pi(5)) = (4, 5, 2, 3, 1)$, *the subsequence* $(4, 5, 3)$ *occupying positions* $1, 2$ *and* 4 *is of type* $\tau = 231$. *The subsequence* $(3, 5, 4)$ *is correspondingly a subsequence of type* $\tau^{\mathsf{l}} = 132$ *of the permutation* $\pi^{\mathsf{l}} = (1, 3, 2, 5, 4)$. *The subsequence occupying positions* $1, 2$ *and* 4 *of* $\pi^- = (2, 1, 4, 3, 5)$ *is of type* $\tau^- = 213$. *Finally, the subsequence* $(4, 1, 2)$ *occupying positions* $3, 4$ *and* 5 *of* $\pi^{-1} = (5, 3, 4, 1, 2)$ *is of type* $\tau^{-1} = 312$.

Since π *avoids the subsequences* 123, 132, *and* 213, *it follows that* π^{l} *avoids* 321, 231, *and* 312, *that* π^- *avoids* 321, 312 *and* 231, *and that* π^{-1} *avoids* 123, 132 *and* 213.

In the following chapter, we will restrict our attention to the case of forbidden subsequences τ of length three. This is the case considered at length by Simion and Schmidt in [18]. We reiterate here some of the material of that paper; this will be generalized in chapter 3. We also introduce some new ideas.

For subsequences of length $k = 3$, the elementary considerations of lemmas 1.2.2 and 1.2.3 provide that $|S_n(123)| = |S_n(321)|$. Similarly, $|S_n(231)|$ is equal to $|S_n(132)|$ by reversal, to $|S_n(213)|$ by complementation, and to $|S_n(312)|$ by inversion. It follows that the enumerative problem for forbidden subsequences of length three is reduced from six to just two cases. We need only choose one exemplar of each of the classes $\{123, 321\}$ and $\{231, 132, 213, 312\}$. In the first two sections, we count the members of $S_n(123)$ and $S_n(231)$. The enumeration of $S_n(123)$ is found by MacMahon [11], Vol. 1, page 130; see also Knuth [7], page 64. The members of $S_n(123)$ are precisely the permutations with no ascending subsequence of length 3. A subsequence of type 231 is often referred to as a *wedge*, and a 231-avoiding permutation is called *wedge-free*.

2 Correspondences between $S_n(123)$ and $S_n(132)$

"Hearing the drums of Catalan beating."
– The Sugarcubes

2.1 Enumeration of $S_n(231)$

We begin our study of forbidden subsequences of length 3 by finding a recurrence relation satisfied by the numbers $|S_n(231)|$.

Assume by induction that we have enumerated $|S_m(231)| = c_m$ for $m < n$, and consider an arbitrary permutation $\pi \in S_n(231)$. Let j be the position such that $\pi(j) = n$. Then the substring $\pi^L = (\pi(1), \pi(2), \ldots, \pi(j-1))$ must consist of the elements $(1, 2, \ldots, j-1)$. For if not, it must contain some element $\pi(i) \geq j$, while the substring $\pi^R = (\pi(j+1), \pi(j+2), \ldots, \pi(n))$ would contain some $\pi(k) < j$. But then we would have a forbidden triple $i < j < k$ with $\pi(k) < \pi(i) < \pi(j) = n$. This is not the case if $\pi \in S_n(231)$.

So the elements of the left substring and the right substring are determined by the position of n. But the permutations π^L and π^R, being subsequences of π, must themselves avoid 231. It is also sufficient that they do so, since if all the elements of π^L are less than all those of π^R there cannot be any subsequence of type 231 with elements in both the left and right substrings. But since an admissible left substring is just an element of $S_{j-1}(231)$, and the admissible right substrings are permutations sequences likewise counted by $|S_{n-j}(231)$, we can invoke the induction hypothesis.

Using the induction hypothesis and summing over j, we thus establish that

$$|S_n(231)| = |S_{n-1}(231)| + \sum_{j=2}^{n-1} |S_{j-1}(231)||S_{n-j}(231)| + |S_{n-1}(231)| = \sum_{j=1}^{n} c_{j-1}c_{n-j}, \quad (1)$$

where we set $c_0 = 1$.

At this point, we might recognize this as the famous recurrence relation for the Catalan numbers, and produce its solution from the literature. It is instructive to give a combinatorial solution, however.

Consider a sequence of length $2n$ composed of n copies of each of the symbols '(' and ')'. Such a sequence of parentheses will be *well-formed* if each '(' can be associated uniquely with a ')' to form nested pairs of parentheses. We will refer to a well-formed sequence of parentheses as a *bracketing sequence*. If a bracketing sequence consists of n open and n closed parentheses, we will say it has length n. Let the set of all bracketing sequences of length n

be denoted by B_n. The members of B_n are elsewhere called *ballot sequences*; this alternative notation emphasizes the characterization that closed parentheses never outnumber open ones as we read from left to right. With a little effort, these two characterizatiosn can be seen to be equivalent

Clearly, the first element of a bracketing sequence must be an open parenthesis; otherwise if it were a ')' it would not have a mate and the sequence would not be well-formed. Consider the ')' which is the mate of the initial open '('. This splits the bracketing sequence into two shorter sequences, each of which must also be well-formed. If the first of these, which falls within the selected pair of brackets, has length $j-1$, then the second, which falls after the selected ')', has length $n-j$. The bracketing sequences therefore satisfy the same recurrence as the 231-avoiding permutations, 1. The enumerative problem of counting the permutations of $S_n(231)$ is thus equivalent to that of counting bracketing sequences of length n.

To enumerate the well-formed sequences of n open and n closed parentheses, observe that there is a bijection between *all* sequences of $n-1$ open and $n+1$ closed parentheses and those sequences of n open and n closed parentheses which are *not* well-formed. If a sequence is not well-formed, there must be a leftmost occurence of a ')' which has equally many '('s and ')'s to its left. Replace each '(' to the right of this by a ')' and each ')' to its right by a '('. We thus obtain a sequence of $n-1$ open and $n+1$ closed parentheses. Similarly, given a sequence of $n-1$ open and $n+1$ closed parentheses, there must be a leftmost ')' which has as many '('s as ')'s to its left. Invert each parenthesis to the right of this location to obtain a sequence with n of each type of symbol, which is not well-formed.

The number of well-formed sequences of n open and n closed parentheses is simply the total number of sequences of n '('s and n ')'s, less the number of such sequences which are not well-formed. This number is

$$c_n = \binom{2n}{n} - \binom{2n}{n-1} = \frac{1}{n+1}\binom{2n}{n}. \tag{2}$$

The numbers c_n are the well-known *Catalan numbers*, named for Eugene Catalan, who co-authored a series of papers on them in 1838 [2]. Catalan himself called them Segner numbers; Johann Andreas von Segner's paper appeared in 1758, with a commentary by Leonhard Euler.

That the numbers 2 satisfy the recurrence 1 is established in an elegant 1838 paper by Lamé [9] which considers triangulations of a polygon. Catalan's paper, which accompanies Lamé's, puts them in the modern form of equation 2 and relates them to, among other things, Legendre polynomials. Segner was, incidentally, also interested in triangulations; his results were generalized by Fuss in 1795. The enumeration of bracketings by considering not-well-formed sequences was presented by D. André in 1878.

We thus have a very simple formula for the number of 231-avoiding permutations of length n.

We can establish a correspondence between 231-avoiding permutations and bracketing sequences in another, intriguing, way. Consider the problem of sorting on a first-in, last-out stack. Given a permutation $\pi = (p_1, p_2, ..., p_n)$, we begin by placing p_1 on the stack. At each step we have the option of adding an element to the stack or removing one from the stack to the final output. We require that every p_i pass through the stack and if $i < j$ we require that p_i be added to the stack before p_j.

Then call a permutation $\pi \in S_n$ *stack sortable* if it can be passed through a stack so that the elements are removed in ascending order. If a given permutation can be sorted, then there is a unique procedure for sorting it: if the next element to be added to the stack is larger than the element on top of the stack, remove the top element from the stack; if it is smaller, add it to the stack; if the stack is empty, add to it; if p_n has been added clear the stack. Consider the sequence of operations which must be performed to sort a given permutation, writing '(' if an element is added to the stack and ')' if one is removed.

Then we have a sequence of n open and n closed parentheses, since each p_i must be added to the stack once and removed once. Also this sequence must be well-formed as defined above, because we can never remove more elements from the stack than have been added to it. This means that, working from left to right, there will always be a surplus of open parentheses, so whenever we encounter a ')' we will be able to supply it with a mate somewhere to its left. We have already determined that the number of such well-formed sequences is the Catalan numbers.

But we claim that the stack-sortable permutations are precisely those which have no subsequences of type 231 (wedges), a result attributed to Knuth [7].

Lemma 2.1.1 *A permutation π is stack-sortable iff $\pi \in S(231)$.*

Proof: If $i < j$ and $\pi(i) < \pi(j)$ then $\pi(i)$ must be removed from the stack before $\pi(j)$ is put on. If $i < k$ and $\pi(i) > \pi(k)$ then $\pi(i)$ must be remain on the stack until after $\pi(k)$ is added. So if $i < j < k$ and $\pi(k) < \pi(i) < \pi(j)$, $\pi(i)$ must be removed before the addition of $\pi(j)$ but after that of $\pi(k)$. But this is impossible, as $\pi(j)$ must be added Before $\pi(k)$. So a stack-sortable permutation cannot have a subsequence of type 231. Conversely, if a permutation avoids 231, it can be sorted according to the algorithm above. The algorithm will fail to sort only if it forces us to remove an element from the top of the stack which is not the largest element which has yet to be removed. Then the top element of the stack is smaller than the next element to be added, but larger than some later element. These three elements constitute a wedge. □

Example 2.1.2 *The bracketing sequence which is associated to a* 231*-avoiding permutation can be found by application of the sorting algorithm.*

For instance, the permutation $(1,3,2,5,4)$ *corresponds to the sequence* $()(())(())$.

The unique stack-sortable permutation which is sorted by a given bracketing sequence : most easily found by observing that a matching pair of parentheses correspond to the insertion and removal of the same element from the stack. Label the closed parentheses in order with the integers 1 through n, and then assign the same integers to the matching open parentheses.

Example 2.1.3 *The bracketing sequence* $()(()(()))$ *corresponds to the permutation* $(1,5,2,4,3)$*:*

	1			2			3	4	5
(	)	(	(	)	(	(	)	)	)
1		5	2		4	3			

2.2 Enumeration of $S_n(123)$

To enumerate the elements of $S_n(123)$, we appeal to the Robinson-Schensted bijection, which associates to each permutation in S_n an ordered pair of standard Young tableaux on n elements and having the same shape. It will be useful in the sequel to have reviewed the Robinson-Schensted correspondence in some detail. Omitted proofs can be found in [17].

Definitions 2.2.1 *Given* m *positive integers* $r_1 \geq r_2 \geq \cdots \geq r_m$*, a standard Young tableau of shape* $(r_1, r_2, \ldots, r_m)$ *is an array of* $r_1 + r_2 + \ldots + r_m$ *distinct positive integers arranged in* m *left-justified rows with* r_j *entries in the* j*th row, with the numbers increasing along each row and down each column.*

The order of the tableau is the total number of entries, $r_1 + r_2 + \ldots + r_m$.

The essential step of the Robinson-Schensted correspondence is given by the following definition and lemma:

Definition 2.2.2 *$Y \leftarrow x$ is the array obtained from the standard tableau Y by following the procedure:*

Step 1 *Insert x in the first row of Y by displacing the smallest number in the row which is larger than x; if no number is larger than x, add x at the end of the row.*

Step 2 *If a number was displaced from the (t)th row, insert this number in the $(t+1)$th row by displacing the smallest number larger than it or by adding it at the end of the row. Repeat this step until some number is added at the end of a row.*

Lemma 2.2.3 *$Y \leftarrow x$ is a standard tableau.*

Proof: omitted. □

The Robinson-Schensted correspondence provides a bijection between the members of S_n and ordered pairs (P, Q) of standard Young tableaux of order n having the same shape. The tableaux created by the following construction are called the P-symbol and the Q-symbol by Craige Schensted, who presented the construction in 1961 [17]. Schensted's work is a more formal and complete presentation of a procedure which was introduced in a different context by Gilbert deB. Robinson in 1938. In the forward direction, the correspondence is defined by the following definition. The steps used to obtain the tableaux P and Q can be inverted to invert the bijection.

Definition 2.2.4 *The P-symbol corresponding to $\pi = (p_1, p_2, \ldots, p_n)$ is the array $(\cdots((p_1 \leftarrow p_2) \leftarrow p_3)\ldots \leftarrow p_n)$.*

The P-symbol is seen to be a standard Young tableau by repeated application of lemma 2.2.3.

Definition 2.2.5 *The Q-symbol corresponding to π is the array obtained by numbering the positions of the shape of P in the order that they are added.*

We omit the proof that the Q-symbol is also a standard Young tableau.

We have omitted the proof that the Robinson-Schensted correspondence actually is a bijection. However, given this, it is evident that each p_j inserted, at its turn, into the r-th box of the first row of P is the largest member of an ascending subsequence of length r in π, the other members of the subsequence being the occupants of the positions to the left of position r at the time p_j is inserted. It is also straightforward to see that there is no longer ascending subsequence of π which claims p_j as its largest member. Following Schensted's notation, we call the set of p_j which are inserted into the r-th box of the first row the *r-th basic subsequence* of π. Each basic subsequence is a decreasing subsequence by the nature of the Robinson-Schensted correspondence.

The permutations we are interested in, the members of $S_n(123)$, are precisely those which have no ascending subsequences of length greater than 2. The Robinson-Schensted correspondence thus associates to each an ordered pair of standard Young tableaux of the same shape, that shape having at most two columns. To this ordered pair (P, Q) we now associate a rectangular standard Young tableau having n rows and 2 columns, in the following manner.

Obtain a new tableau Q' by rotating Q by one half-turn, and replacing each entry x by $2n + 1 - x$. The entries of Q' are $n + 1, n + 2, \ldots, 2n$ and Q' increases along rows and columns. Now join P and Q' to obtain the desired tableau of shape 2 by n; call it T. Clearly, T contains the elements $1, 2, 3, \ldots, 2n$. Furthermore, it increases along rows and columns, since each of the two columns consists first of elements of P, which increase down columns, and then of elements of Q', which increase down columns and are larger. (Similarly for rows.) Therefore, T is a standard Young tableau.

Now we create a sequence of length $2n$ from the symbols '(' and ')', letting the j-th element of the sequence be '(' if j appears in the first column of T, and ')' if in the second column. This sequence will consist of n open parentheses and n closed parentheses. The condition that T be increasing by rows and columns ensures that the sequence of parentheses will be well-formed in the sense that we will never close more parentheses than we have opened; this is the alternative characterization of the well-formed bracketing sequences.

We have already enumerated such bracketing sequences in the previous section, and found the number with n open and n closed parentheses to be the n-th Catalan number,

$$c_n = \binom{2n}{n} - \binom{2n}{n-1} = \frac{1}{n+1}\binom{2n}{n}.$$

More to the point, bracketing sequences of length n are equinumerous with the set $S_n(231)$. The sets $S_n(123)$ and $S_n(231)$ are thus seen to be have the same cardinality. We can state this as a theorem. For the statement of the theorem, we prefer to recall that we have a bijection between 231-avoiding and 132-avoiding permutations; we obtain the permutations of one set by reversing each member of the other. Therefore also $S_n(132) = c_n$. The reason for this choice is that in the following sections we will find alternative bijective correspondences between the two sets $S_n(123)$ and $S_n(132)$.

Theorem 2.2.6 $|S_n(123)| = |S_n(132)| = c_n$.

Proof: $\square$

Recall that the implication of this result is that restricting permutations by forbidding *any* subsequence of length $k = 3$ results in the same distribution.

Corollary 2.2.7 $|S_n(\pi)| = |S_n(\rho)|$ *for any* $\pi, \rho \in S_3$.

Proof: $\square$

The method by which we obtained 2.2.6, while it actually produces an explicit bijection between the two sets by composing the maps which associate a bracketing sequence with the elements of each, appears to be accidental at best. It provides us with no intuition about the existence of analogous bijections for $\pi, \rho \in S_k$ with $k > 3$. This is because it is far from clear what commonalities between the forbidden permutations $(1,2,3)$ and $(1,3,2)$ we have appealed to. In the following three sections we will consider three alternative proofs of 2.2.6.

All three of these proofs, which are strikingly different, seem to exploit different features of the forbidden permutations. One reason to gather together competing proofs of the same theorem is the hope that some insight might be gained from their alternative viewpoints. Another is the hope that each might be open to generalization in a different direction, providing a catalogue of bijections between $S_n(\pi)$ and $S_n(\rho)$ for some selections of π and ρ of length greater than three.

We will obtain some such generalizations in chapter three. It is natural to wonder whether we will be able to conclude that $|S_n(\pi)| = |S_n(\rho)|$ in all cases. We will also see in chapter three that this is quite strongly not the case.

2.3 The Simion-Schmidt bijection

Recall that the arguments of the previous two sections do provide one bijection between $S_n(123)$ and $S_n(132)$, via well-formed sequences of parentheses. In addition to being cumbersome, it has the disadvantage of failing to fix the intersection $S_n(123) \cap S_n(132)$, even as a set. Rodica Simion and Frank Schmidt [18] introduced a simpler bijective correspondence which fixes each element of $S_n(123) \cap S_n(132)$. Following their approach, we describe the correspondence through a pair of mutually inverse algorithms.

Algorithm 2.3.1

Input: $\sigma = (a_1, a_2, ..., a_n) \in S_n(123)$

Output: $\tau = (c_1, c_2, ..., c_n) \in S_n(132)$

Step 1: $i \leftarrow 1;$

Step 2: $c_i \leftarrow a_i,\ x \leftarrow a_i;$

Step 3: $i \leftarrow i + 1;$ *if* $i > n$, *exit;*

Step 4: *if* $x > a_i$, *then* $c_i \leftarrow a_i$, *go to (3);*
else, $c_i \leftarrow \min\{k | x < k \leq n, k \neq c_j \text{ for all } j < i\}$, *go to (3).*

Algorithm 2.3.2

Input: $\tau = (c_1, c_2, ..., c_n) \in S_n(132)$

Output: $\sigma = (a_1, a_2, ..., a_n) \in S_n(123)$

Step 1: $i \leftarrow 1;$

Step 2: $a_i \leftarrow c_i, x \leftarrow c + i;$

Step 3: $i \leftarrow i + 1;$ *if* $i > n$, *exit;*

Step 4: *if* $x > c_i$, *then* $a_i \leftarrow c_i, x \leftarrow c_i$, *go to (3);*
else, $a_i \leftarrow \max\{k | k \leq n, k \neq a_j \text{ for all } j < i\}$, *go to (3).*

It is easy to verify that the two constructions produce output of the desired type, by considering the fourth step in each case. It is likewise easy to see that the algorithms invert one another, and thus that a bijection has been produced. Also notice that if a permutation is both 123-avoiding and 132-avoiding that it will be invariant under either algorithm.

The fourth step in algorithm 2.3.1 involves a branch, and thus divides the elements of a 123-avoiding permutation into two types, those for which $x > a_i$ and those for which $x < a_i$. These are precisely the two basic subsequences of the Robinson-Schensted correspondence. Algorithm 2.3.1 fixes the first basic subsequence. If the permutation σ is also 132-avoiding, then the second basic subsequence will be fixed as well, but in general this is not the case.

We can consider the algorithms as determining analogues to the basic subsequences in the context of 132-avoiding permutations. The first basic subsequence consists of just the *left-to-right minima* of the permutation, as is the case in the 123-avoiding case. Under this definition, we can execute either algorithm by first identifying and fixing the first basic subsequence, and second rearranging the elements of the second basic subsequence. In algorithm 2.3.2 they are arranged into descending order; in algorithm 2.3.1 they are arranged into ascending order, subject to the restriction of not introducing any new left-to-right-minima.

Example 2.3.3 *The algorithms 2.3.1 and 2.3.2, applied to all inputs for* $n = 3$, *give the following correspondence. On the left are the* 123*-avoiding permutations of length 3. On the right are the* 132*-avoiding permutations. The middle column shows the members of the first basic subsequence of the permutations on the left, and likewise of the analogous structure in the right permutations. Notice how these are fixed by the correspondence.*

$$\begin{array}{lclcl}
132 & & 1 & & 123 \\
213 & \longleftrightarrow & 2\ 1\ \ \ & \longleftrightarrow & 213 \\
231 & & 2\ \ \ \ 1 & & 231 \\
312 & \longleftrightarrow & 3\ 1\ \ \ & \longleftrightarrow & 312 \\
321 & & 3\ 2\ 1 & & 321
\end{array}$$

Notice also how all permutations except those in the top row are unaltered by the bijection.

2.4 The Richards bijection

In this section we consider an bijection introduced by Dana Richards [14] between the permutations $S_n(123)$ and the bracketing sequences of B_n. (Richards refers to the elements of B_n as ballot sequences and writes them with the symbols '0' and '1'.) We can combine this with the results of 2.1 to obtain a bijection between $S_n(123)$ and $S_n(132)$. The resulting correspondence is different from that of the previous section, and fails to fix the intersection of the two sets.

Like Simion and Schmidt, Richards presents the bijection in the form of two mutually inverse algorithms. We preserve that approach.

Algorithm 2.4.1

Input: $\beta = (b_1, b_2, \ldots, b_{2n}) \in B_n$

Output: $\pi = (p_1, p_2, \ldots, p_n) \in S_n(123)$

Step 1: *Initialize two cursors,* $r \leftarrow n+1$, $c \leftarrow n+1$;

Step 2: *Set* $j \leftarrow 1$;

Step 3: *Repeat steps (4) and (5) for* $i = 1$ *to* n;

Step 4: *If* $b_j =$ '(' *do step (C), else do step (R);*

Step 5: *Increment* $j \leftarrow j + 1$.

Step C: *Repeat*

$$c \leftarrow c - 1; \quad j \leftarrow j + 1$$

until $b_j =$ ')'*;*
set $p_c \leftarrow i$

Step R: *Repeat*

$$r \leftarrow r - 1$$

until p_r *is unset;*
set $p_r \leftarrow i$.

The operation of the algorithm inserts the elements $1, 2 \ldots n$ in order. Each insertion is made as one of the n ')'s is read from the bracketing sequence. The elements are divided into two types: those inserted at the position of cursor c in step C, and those inserted at the postion of cursor r in step R. These can be seen as two subsequences of the permutation, C and R. The second type of entry, comprising subsequence R, is always inserted at the rightmost available position. As each cursor moves always to the left, each of the two types forms a decreasing subsequence in the final permutation.

That the algorithm produces a valid permutation is a consequence of the fundamental property of bracketing sequences. Cursor c moves one step to the left every time a '(' is processesed. An entry is made in the permutation whenever a ')' is processed. Since we never reach a point where we have processed more closed than open parentheses, there are always unset entries at or to the right of cursor c. When the algorithm terminates, n open parentheses have been processed, cursor c has moved all the way to the left, and all positions are available to be filled.

That the permutation produced is 123-avoiding follows from the fact that the two subsequences C and R decrease from left to right. Any ascending subsequence of length three would have to have at least two entries in one or the other subsequence. In fact, in the terminology of section 2.2, the subsequence C is just the first basic subsequence, namely the set of left-to-right minima of π. The subsequence R is the second basic subsequence. Let us state this as a proposition.

Proposition 2.4.2 *The elements entered in step C of algorithm 2.4.1 constitute the first basic subsequence of the permutation π. The elements entered at step R form the second basic subsequence of π.*

Proof: Suppose the element k is being entered into position p_c at step C of the algorithm. Then we are processing the k-th closed parenthesis from the sequence β. Since β is well-formed, at least k open parentheses have already been processed, so cursor c has moved at least k positions from the right end of the sequence. But only $k-1$ entries have already been made, namely all those smaller than element k. Each has been made either at step C, when cursor c was to the right of where it is now, or at step R, in which case it was placed in the rightmost available position. In either case, it was placed to the right of the current position of c. So there are no elements smaller than k to the left of k; k is therefore a left-to-right minimum. On the other hand, suppose element l is being entered into position p_r at step R. Since β is well-formed, cursor c has already moved l steps to the left. The rightmost available position is therefore to the right of c. But since we are at step R, an entry has already been made at c. Since this entry was made before l, it is smaller than l, and it is to the left of l. Therefore, l is not a left-to-right minimum. □

So, like the Simion-Schmidt algorithm, the Richards algorithm makes strong use of the fact that a 123-avoiding permutation is composed of two interleaved, descending, basic subsequences. We give next the inverse algorithm, which converts a 123-avoiding permutation to a bracketing sequence.

Algorithm 2.4.3

Input: $\pi = (p_1, p_2, \ldots, p_n) \in S_n(123)$

Output: $\beta = (b_1, b_2, \ldots, b_{2n}) \in B_n$

Step 1: *Initialize an array, seen*$[1 \ldots n] \leftarrow$ *false;*

Step 2: *Initialize a cursor,* $c \leftarrow n + 1$;

Step 3: $j \leftarrow 1$;

Step 4: *For* $i = 1$ *to* n *do (5) and (6);*

Step 5: *If* i *has been seen, do (S); else do (NS).*

Step 6: *Increment* $j \leftarrow j + 1$.

Step S: *Set* $b_j \leftarrow$ ')'

Step NS: *Repeat*

$c \leftarrow c - 1$;
$b_j \leftarrow$ '(';
$j \leftarrow j + 1$;
set p_c *as seen*

until $p_c = i$;
set $b_j \leftarrow$ ')'

The algorithm searches, right-to-left, for the smallest element which has not yet been seen. In other words, it looks for the left-to-right minima, starting with the smallest and rightmost. If it has to move t elements to the left to reach the next such element, t open parentheses are added at step NS. If an element l is passed over while scanning for a smaller one, in other words if l is not a left-to-right minimum, then an open parenthesis is added for l when the counter i reaches l, and l is not searched for.

It should be clear that algorithm 2.4.3 simply inverts the process of algorithm 2.4.1.

Example 2.4.4 *The bracketing sequence* ()((())()) *is well-formed. We carry out algorithm 2.4.1 on this input. The positions of the cursors are underscored at each step.*

step	*processed input*	*output*
0		$(x, x, x, x, x)_{=}$
1	()	$(x, x, x, x, \underline{1})\_$
2	()((()	$(x, \underline{2}, x, x, 1)\_$
3	()((())	$(x, \underline{2}, x, \underline{3}, 1)$
4	()((())()	$(\underline{4}, 2, x, \underline{3}, 1)$
5	()((())())	$(\underline{4}, 2, \underline{5}, 3, 1)$

We next carry out the inverse algorithm 2.4.3 on the resulting permutation $(4, 2, 5, 3, 1)$.

step	*cursor*	*seen*	*output*
0	$(4, 2, 5, 3, 1)\_$		
1	$(4, 2, 5, 3, \underline{1})$	1	()
2	$(4, \underline{2}, 5, 3, 1)$	1, 2, 3, 5	()((()
3	$(4, \underline{2}, 5, 3, 1)$	1, 2, 3, 5	()((())
4	$(\underline{4}, 2, 5, 3, 1)$	1, 2, 3, 4, 5	()((())()
5	$(\underline{4}, 2, 5, 3, 1)$	1, 2, 3, 4, 5	()((())())

Next we present a recursive algorithm for converting a bracketing sequence to a 132-avoiding permutation. We make use of the way in which each set satisfies the Catalan recurrence 1.

Algorithm 2.4.5

Input: $\beta = (b_1, b_2, \ldots, b_{2n}) \in B_n$;
a set $Z = x_1 < x_2 < \cdots < x_n$ *of* n *integers*

Output: *A 132-avoiding permutation sequence* $\pi = (p_1, \ldots, p_n)$

Step 1: *Find the '$)$' matching* $b_1 =$ '$($' *by scanning from left to right for the first position at which the numbers of '$($' and '$)$' scanned is equal. Suppose this is position* $2i$;

Step 2: *Let* $p_i \leftarrow \max\{z : z \in Z\}$;

Step 3: *Fill positions* $1, 2, \ldots, (i-1)$ *of* π *with the integers* $Z_{\max} = \{x_{n-i+1}, \ldots, x_{n-1}\}$ *by calling algorithm 2.4.5 with input* $\beta_L = (b_2, b_3, \ldots, b_{i-1})$ *and* $Z_{\max}$;

Step 4: *Fill positions* $(i+1), (i+2), \ldots, n$ *of* π *with the integers* $Z_{\min} = \{x_1, x_2, \ldots, x_{n-i}\}$ *by calling algorithm 2.4.5 with input* $\beta_R = (b_{i+1}, \ldots, b_n)$ *and* $Z_{\min}$.

The inverse algorithm simply reverses the steps of algorithm 2.4.5. We can form an algorithm for converting the permutations of $S_n(123)$ to those of $S_n(132)$ by concatenating the algorithms 2.4.3 and 2.4.5. This algorithm fails to fix the intersection of the two sets, as can be seen from the example below. Thus it provides a differet correspondence from that of the Simion-Schmidt bijection in the previous section.

Example 2.4.6 *The algorithms 2.4.3 and 2.4.5, applied to all inputs for* $n = 3$, *give the following results:*

1 3 2		((()))		123
2 1 3	$\longrightarrow$	(()())	$\longrightarrow$	213
2 3 1		()(())		312
3 1 2	$\longrightarrow$	(())()	$\longrightarrow$	231
3 2 1		()()()		321

Algorithm 2.4.5 can be modified by working from the back, rather than the front, of the sequence β. The reader may verify that the correspondence between $S_n(123)$ and $S_n(132)$ thus produced still differs from the Simion-Schmidt bijection.

2.5 A Catalan tree

2.5.1 In connection with forbidden subsequences

We define recursively a rooted tree whose vertices are identified with the permutations in the set $S(123)$. Let the root be the permutation $(1) \in S_1(123)$, and let each $\pi \in S_n(123)$ be a child of the permutation $\pi' \in S_{n-1}(123)$ obtained from π by deleting the largest element, n. (Clearly, deleting elements from a permutation cannot introduce any forbidden subsequences.) Call this tree $T(123)$. It will also be convenient to label each vertex of $T(123)$ with the number of its children. Then the structure of the tree $T(123)$ is characterized by the following theorem.

Lemma 2.5.1 *If $\pi \in T(123)$ has the label t, then its t children are labelled* $2, 3, \ldots, t, t+1$.

Proof: The number of children of a permutation $\pi = (p_1, p_2, \ldots, p_n)$ is the number of sites at which $n+1$ may be inserted without introducing a subsequence of type 123. If the leftmost element of π which is not a left-to-right minimum of π is p_j, then $n+1$ may be inserted anywhere to the left of p_j, but nowhere to its right. Thus if π has the label t as a vertex of $T(123)$, then p_t is the leftmost element of the second basic subsequence of π. (Possibly $t = n+1$, in which case π is the descending permutation.)

If $n+1$ is inserted in the leftmost site, the new permutation $\pi^* = (n+1, p_1, p_2, \ldots, p_n)$ has $t+1$ sites where $n+2$ may legitimately be inserted, namely all those to the left of p_t. Therefore π^* receives the label $t+1$.

On the other hand, if n is inserted elsewhere to the left of p_t, say in position s to form π^s, n itself becomes the leftmost member of the second basic subsequence of π^s. Hence π^s receives the label s.

Since the children of π in $T(123)$ will be $\pi^2, \pi^3, \ldots, \pi^t$ and π^*, and since these receive the labels $2, 3, \ldots, t, t+1$ respectively, the proof is complete. □

We define a second tree with vertex set $S(132)$ analogously, and call it $T(132)$. The structure theorem for the tree $T(132)$ is identical to the one for $T(123)$, namely:

Lemma 2.5.2 *If $\pi \in T(132)$ has the label t, then its t children are labelled* $2, 3, \ldots, t, t+1$.

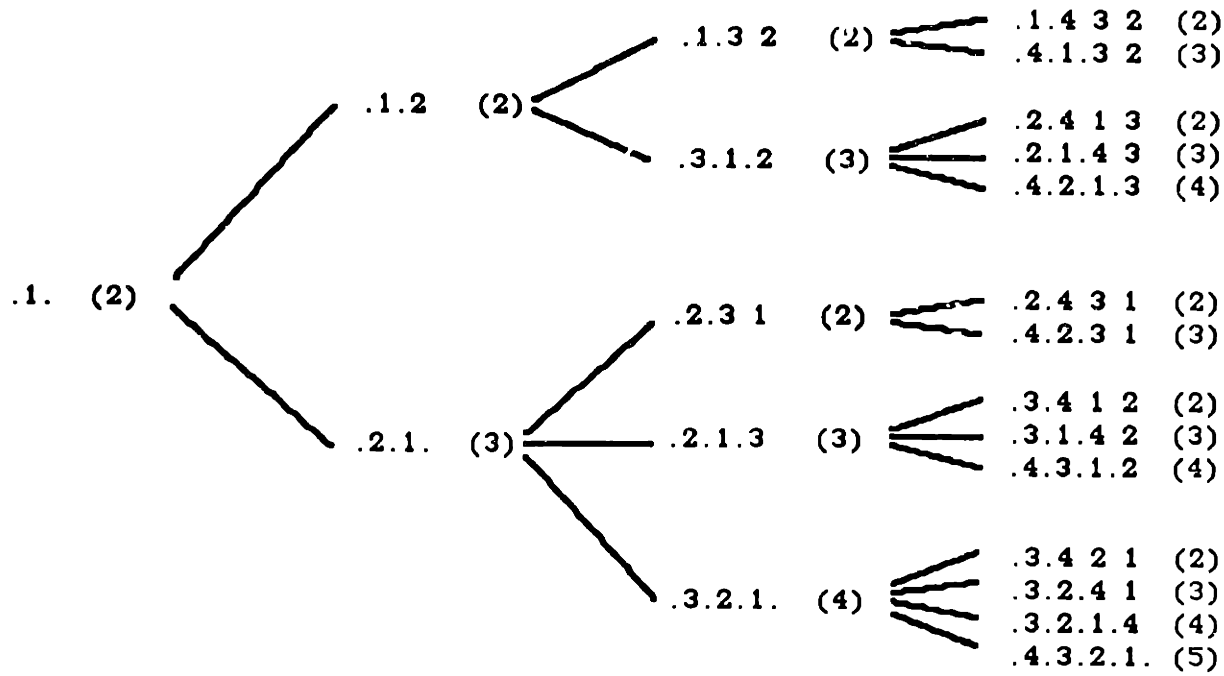
Proof: Suppose $\pi = (p_1, p_2, \ldots, p_n) \in T(132)$ has the label t. Then there are t sites in π where $n+1$ may be inserted without introducing a forbidden subsequence of type 132.

One of these sites will always be the leftmost position, so first suppose $n+1$ is inserted at the left of π to form $\pi^* = (n+1, p_1, p_2, \ldots, p_n)$. Then π^* also has t sites where is is possible to insert $n+2$ to the right of $n+1$. So π^* receives the label $t+1$.

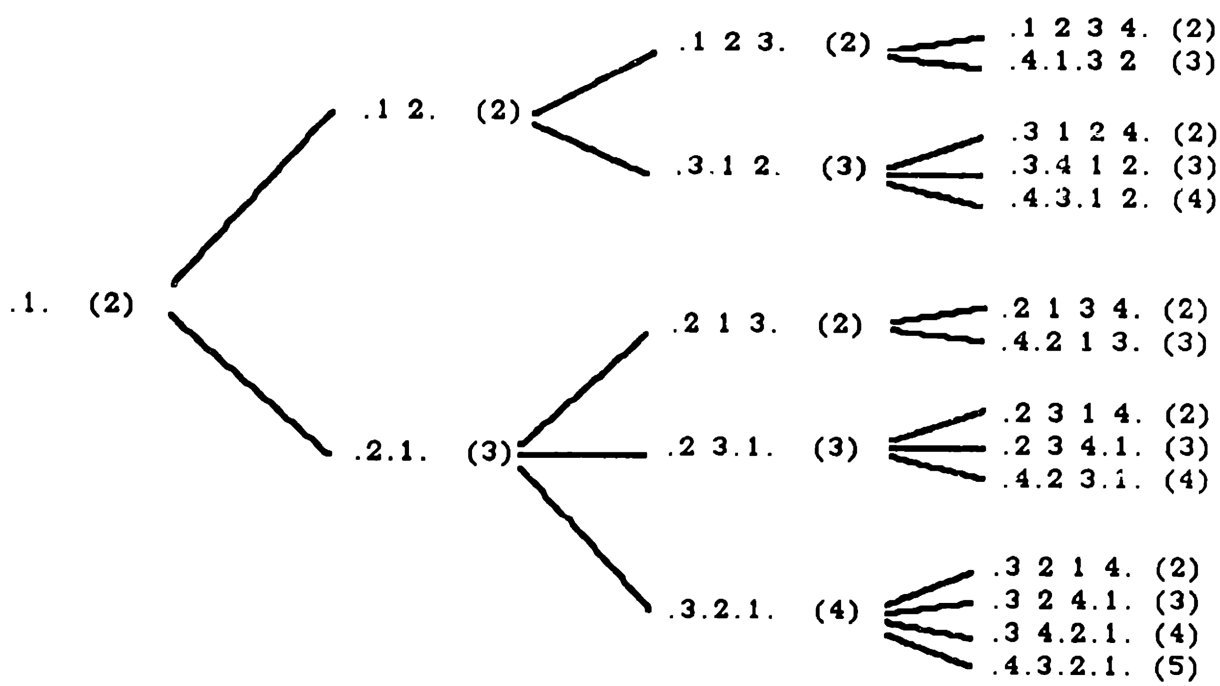
Alternatively, for $s = 2, 3, \ldots t$, suppose $n+1$ is inserted into the s-th site to obtain π^s. Then $n+2$ cannot be inserted anywhere to the left of $n+1$ (except to the left of p_1) because the subsequence consisting of the elements $(p_1, n+2, n+1)$ would be of the forbidden type 132. However, each of the $t-(s-1)$ sites to the right of $n+1$ is still available, in addition to the leftmost position, so π^s must be labelled $t-s+2$.

Since the children of π in $T(132)$ will beπ^*, π^2, π^3, $\ldots \pi^t$, and since these are labelled $t+1, t, \ldots, 3, 2$ respectively, the proof is complete. □

In addition to completing a fourth proof of 2.2.6, we have actually proven that $T(123)$ and $T(132)$ are isomorphic trees with number of vertices on the n-th level being c_n, the Catalan number. Since the children of any vertex receive distinct labels, a vertex can be uniquely determined in each of the trees by listing the labels of its ancestors along a path back to the root. Therefore there is a unique isomorphism between $T(123)$ and $T(132)$, and since the vertices on the n-th level of $T(123)$ are identified with the members of $S_n(123)$ (and similarly for $T(132)$), this induces a bijection between $S_n(123)$ and $S_n(132)$ for every n.



The tree T(123) with labels showing active sites



The isomorphic tree T(132)

Note that this is not the same as the Simion-Schmidt bijection. The vertex associated with the permutation $(2,3,1)$ in each tree corresponds to $(2,1,3)$ in the other. Since $(2,3,1) \in S_n(123) \cap S_n(132)$, it must be fixed by the Simion-Schmidt correspondence.

Example 2.5.3 *In each line of the following table we list first an element $\pi \in S_3(123)$; in the middle, the corresponding labels of the ancestors of the node associated to π in $T(123)$, including the node itself; on the right, the permutation associated with the node of $T(132)$ reached by following the path with the same labels. It is evident from comparing this small example with 2.3.3 and 2.4.6 that the bijection induced by the tree isomorphism differs from those of the previous sections.*

132	$(2,2,2)$	123
312	$(2,2,3)$	312
231	$(2,3,2)$	213
213	$(2,3,3)$	231
321	$(2,3,4)$	321

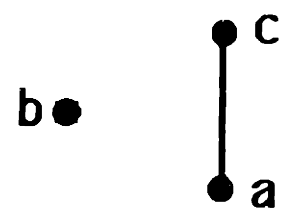
If a sequence of vertex labels $(f_1, f_2, \dots, f_n)$ having the property that $2 \leq f_i \leq f_{i-1}+1$ is converted into a sequence $(a_1, a_2, \dots, a_n)$ according to $a_i = i+2-f_i$, then the new sequence will be non-decreasing with each $a_i \leq i$. Such sequences are a familiar instance of the Catalan numbers, being naturally associated with lattice paths.

It is straightforward to convert the observations underlying theorems 2.5.1 and 2.5.2 to algorithms for transforming 123-avoiding permutations into 132-avoiding permutations and vice versa.

2.5.2 In relation to some minimal semiorders

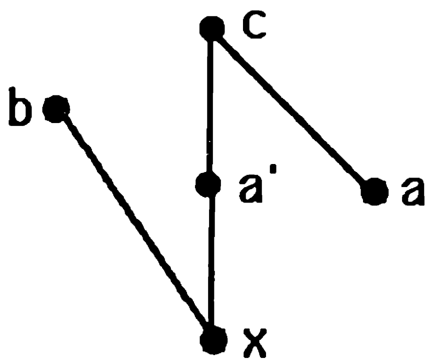
As a final observation, the minimal semiorders introduced by Karen Stellpflug in [20] are also in correspondence with the nodes of the trees defined above. A partially ordered set is a semiorder if and only if it can be represented by a set of equal length open intervals in the real line, with the order relation $(a,b) < (c,d)$ if and only if $b \leq c$. A semiorder has representation number k if it has a representation in which all intervals have the same integer length k, but has no such representation with intervals of length $k-1$.

A semiorder has representation number 1 if and only if it has no suborder isomorphic to the following order, which Stellpflug calls $S_{2,1}$:

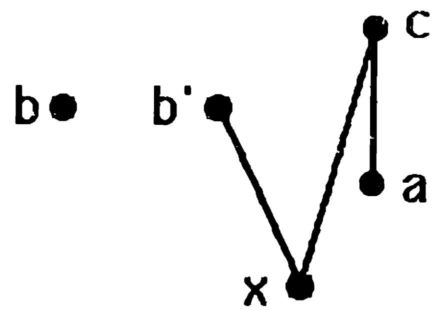


The semiorder $S_{2,1}$

Likewise, a semiorder has representation number ≤ 2 if and only if it has no suborder isomorphic to either of the following semiorders of representation number 3:



The semiorder $S_{3,1}$



The semiorder $S_{3,2}$

Stellpflug show how to obtain the minimal 3-representable semiorders from the minimal 2-representable semiorder by a process of duplicating a minimal element, and then adding a new minimal element which distinguishes the minimal node from the duplicate. The new minimal node is made incomparable to the original, but placed below the duplicate, and the resulting order is extended to be a semiorder. It turns out that the unique way to do so is to place the new minimal element also below all elements which follow the duplicate in the predecessor extension of the semiorder. If a minimal k-representable semiorder has r minimal elements, then it will give rise to r minimal $k+1$-representable semiorders, by duplicating each of the minimal elements in turn. Because of the way in which the semiorder was constructed, these $k+1$-representable semiorders have variously $2, 3, 4, \dots, r+1$ minimal elements.

Hence, Stellpflug's minimal k-representable semiorders can be seen to correspond to the vertices on the kth level of the Catalan trees examined above. The tree edges correspond to Stellpflug's splitting operator for construction of a minimal semiorder As a direct consequence, Stellpflug's minimal k-representable semiorders are counted by the k-th Catalan number. Stellpflug reports an alternate solution of this same result.

Interestingly, the total number of n-element semiorders is also given by c_n.

3 What is known for $k > 3$

"...ma diceva molto sul furore combinatorio con cui Belbo si era avvicinato alla macchina."
– Umberto Eco, Il Pendolo di Foucault 3

3.1 Generalizations of the Catalan trees

3.1.1 Three trees for $k = 4$

Let a *site* of a permutation

$$\pi = (p_1, p_2, \ldots, p_n) \tag{3}$$

be one of the $n+1$ locations preceding, between, or following the elements of π. Then to *insert n+1 into site i* is to form the permutation

$$\pi^i = (p_1, p_2, \ldots, p_{i-1}, n+1, p_i, p_{i+1}, \ldots, p_n). \tag{4}$$

Definition 3.1.1 *With respect to a particular forbidden subpattern τ, we will call site i of a permutation $\pi \in S(\tau)$ an* active site *if the insertion of $n+1$ into site i creates a permutation $\pi^i \in S(\tau)$.*

For definiteness, we take throughout this section $\tau \in S_4$.

Evidently, if site i is not active then the insertion of $n+1$ into site i gives rise to a subsequence of type τ, $(p_{a(1)}, p_{a(2)}, p_{a(3)}, p_{a(4)})$ where $a(1) < a(2) < a(3) < a(4)$, and $p_{a(k)} = n+1$ iff $t_k = 4$.

Define recursively a rooted tree, $T(\tau)$ whose vertices are identified with the permutations of the set $S(\tau)$. Let the root be the permutation $(1) \in S_1(\tau)$, and let each $\pi \in S_{n+1}(\tau)$ be a child of the permutation $\pi' \in S_n(\tau)$ obtained from π by deleting the largest element, $n+1$.

The number of children of a node π of $T(\tau)$ is the number of active sites of π with respect to τ.

First consider the tree $T(1234)$. To each permutation $\pi \in S_n(1234)$, regarded as a node of $T(1234)$, we associate an ordered pair (x, y) as follows. Let x be one greater than the length of the longest initial decreasing subsequence of π. That is, x is the index of the first element of the second basic subsequence, if one exists, and $n+1$ otherwise. Let y be the number of active sites in π. In this instance, y is the index of the first element of the third basic subsequence of π, if one exists, and $n+1$ otherwise. As a node of $T(1234)$, π has y children. We prove the following theorem.

Theorem 3.1.2 *If $\pi \in S_n(1234)$ is associated with the ordered pair (x, y) in the tree $T(1234)$, then the y children of π in $T(1234)$ are associated with the ordered pairs*

$$(2, y+1), (3, y+1), \ldots, (x, y+1), (x, x+1), (x, x+2), \ldots, (x, y), (x+1, y+1). \quad (5)$$

Proof: The y active sites of π are the first y sites. We verify that inserting $n+1$ into site i gives rise to a permutation π^i associated in $T(1234)$ with the ordered pair

$$\begin{array}{ll} (x+1, y+1) & \text{if } i = 1, \\ (i, y+1) & \text{if } 2 \leq i \leq x, \\ (x, i) & \text{if } x+1 \leq i \leq y. \end{array}$$

If $i = 1$, $n+1$ is inserted to the left of all elements of the permutation. The elements of π with were the leftmost members of the second and third basic subsequences remain the leftmost members of their respective subsequences. Each is displaced one position to the right by the insertion of $n+1$.

If $2 \leq i \leq x$, then $n+1$ is inserted to the left of the first element of the second basic subsequence. The element $n+1$ thus itself becomes the leftmost element of this subsequence, being larger than any element to its left, of which there is at least one as $i \geq 2$. On the other hand, the insertion of $n+1$ does not create any new ascending subsequences of length 3, so the leftmost element of the third basic subsequence is unchanged but moved one position to the right.

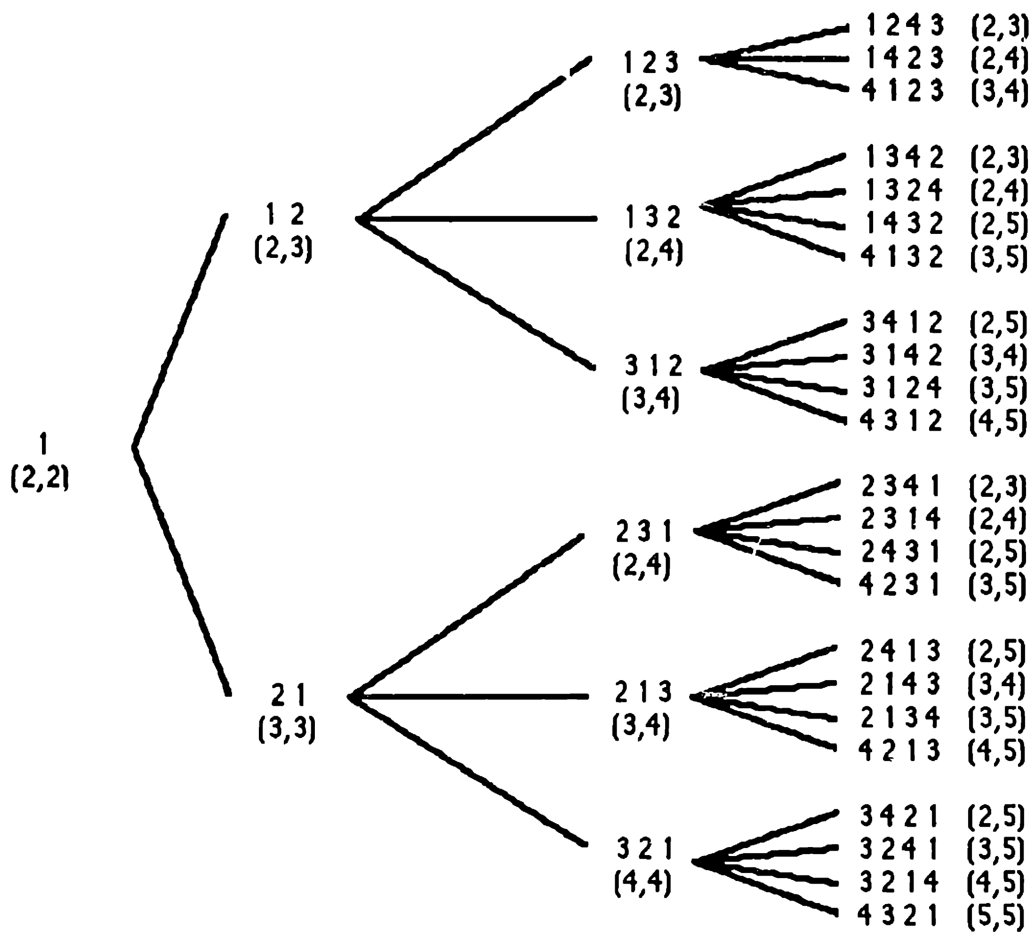
Finally, if $i \geq x+1$, then (p_{x-1}, p_x, p_i) is a subsequence of type 123. (Note that if $i \geq x+1$, it must be the case that $x \leq n$, so that p_x actually exists.) Hence $p_i = n+1$ becomes the new leftmost member of the third basic subsequence. The role of the element p_x is unchanged. $\square$

Example 3.1.3 *Let π be the permutation*

5317462

has initial decreasing sequence of length 3. *Of the* 8 *sites, the leftmost* 6 *are active. Therefore* π *receives the label* $(3,6)$. *Inserting the element* 8 *into each of the* 6 *active sites in turn from left to right produces permutations with the labels*

$$(4,7); (2,7); (3,7); (3,4); (3,5); (3,6)$$



The tree T(1234) with the labels (x,y) at each node

Next consider the tree $T(1243)$. We again associate to each node an ordered pair (x, y). Again let x be one greater than the length of the longest initial decreasing subsequence of π. As in the previous case, $2 \leq x \leq n+1$.

Also, again let y be the number of active sites in π. In the previous case, $\tau = 1234$, the y active sites were the y leftmost sites of π. This is no longer necessarily the case. But it is still true that as a node of $T(1243)$, π has y children.

In the following theorem, we prove that $T(1243)$ has the same structure as $T(1234)$.

Theorem 3.1.4 *If $\pi \in S_n(1243)$ is associated with the ordered pair (x, y) in the tree $T(1243)$, then the y children of π in $T(1243)$ are associated with the ordered pairs*

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

Proof: Let the y active sites be numbered from left to right as the first active site, second active site, ..., ith active site, ...

While the active sites are no longer necessarily the y leftmost sites of π, it is the case that the first x sites are all active. For if $n+1$ is to introduce a subsequence of type 1243, it must fall in the third position of such a subsequence, and to the right of both elements of some increasing subsequence of length two.

The remaining $y - x$ active sites fall to the right of position x.

It is helpful to observe that the introduction of a new element, $n+1$, creates a permutation with one extra site.

The key observation is that if the site located in π as defined in 3 between p_k and p_{k+1} is not active, then neither is the site between p_k and p_{k+1} in π^i as defined in 4. For if the interposition of $n+1$ between p_k and p_{k+1} gives rise to a forbidden subpattern in 3, then the interposition of $n+2$ in the same location gives rise to essentially the same forbidden subsequence.

If the site between p_{i-1} and p_i is active in π, then both of the new sites $(p_{i-1}, n+1)$ and $(n+1, p_i)$ are potentially active in π^i.

On the other hand, we may verify that if a site was active in π, it remains active in π^i unless it falls to the right of position x and to the left of position i, where $n+1$ was inserted.

We thus observe that inserting $n+1$ into the ith active site gives rise to a permutation π^i associated in $T(1243)$ with the ordered pair

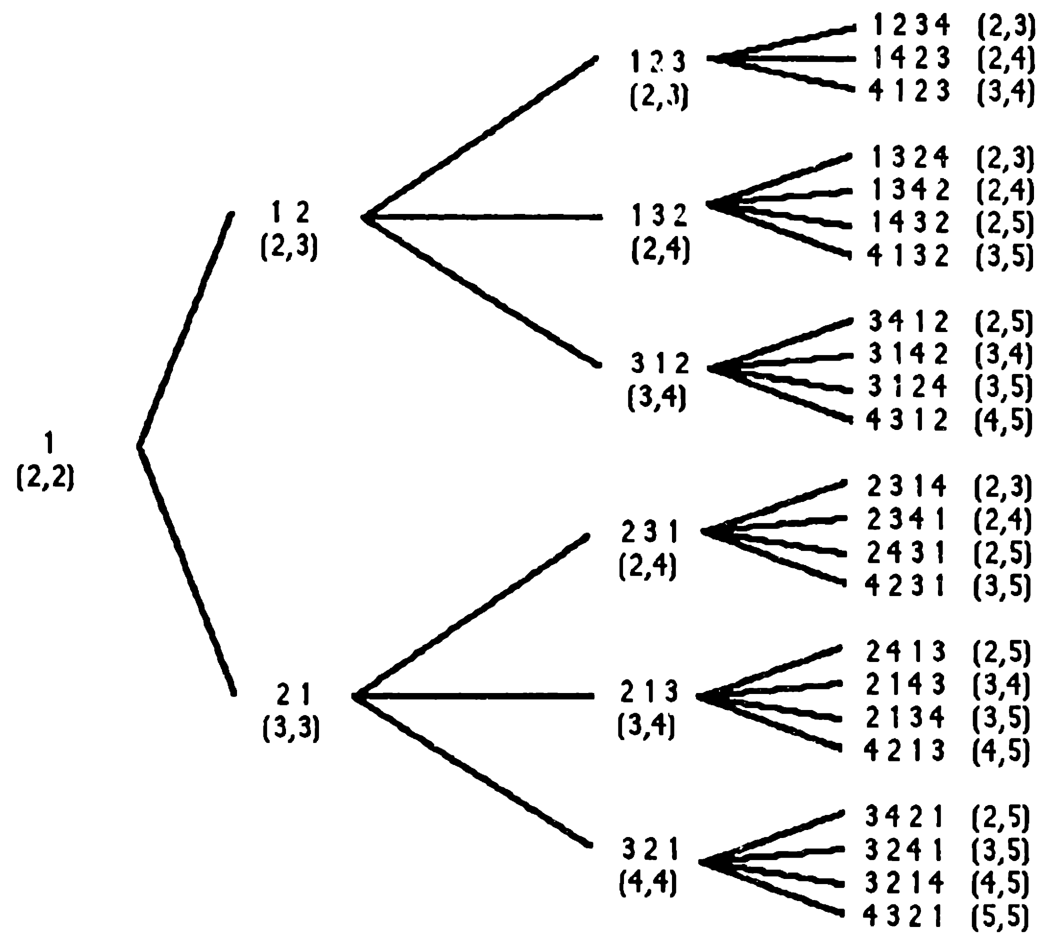
$$\begin{array}{ll} (x+1, y+1) & \text{if } i = 1, \\ (i, y+1) & \text{if } 2 \le i \le x, \\ (x, x+y+1-i) & \text{if } x+1 \le i \le y. \end{array}$$

If $i = 1$, $n+1$ is inserted to the left of all elements of the permutation. This removes no active sites, and creates one new one, namely to the left of $n+1$ itself.

The first x active sites are simply the x leftmost sites. If $2 \le i \le x$, then $n+1$ is inserted to the left of the first position in which there is an left-to-right increase. Since $n+1$

is larger than all other elements of π^i, it is larger than the element to its left, and becomes the new first marker of a left-to-right increase. Also, all active sites of π remain active sites of π_i, and one new one is introduced, to the left of $n+1$.

Finally, if $i \geq x+1$, then $(p_{x-1}, p_x, n+1)$ is a subsequence of type 123. (Note that if $i \geq x+1$, it must be the case that $x \leq n$, so that p_x actually exists.) Then all sites between p_x and $n+1$ become inactive. The sites which remain active are the initial x sites and the final $(y-x+1)-(i-x)$ active sites. (There are $y-x$ active sites to the right of position x in π, plus one extra site formed by the introduction of $n+1$, less the $i-x$ active sites eliminated between p_x and $n+1$.) □



The tree T(1243) with the labels (x,y) at each node

Thirdly, consider the tree $T(2143)$. We again associate to each node an ordered pair (x, y). This time, let x be one greater than the length of the longest initial *increasing* subsequence of π. Again let y be the number of active sites in π, i.e. the number of children of π in $T(2143)$.

In the following theorem, we prove that $T(2143)$ has the same local structure as the two previous trees.

Theorem 3.1.5 *If $\pi \in S_n(2143)$ is associated with the ordered pair (x, y) in the tree $T(2143)$, then the y children of π in $T(2143)$ are associated with the ordered pairs*

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

Proof: Let the y active sites be numbered from left to right as the first active site, second active site, ..., ith active site, ...

It is again case that the first x sites are all active. For if $n+1$ is to introduce a subsequence of type 2143, it must fall in the third position of such a subsequence, and to the right of both elements of some decreasing subsequence of length two.

The remaining $y - x$ active sites fall to the right of position x.

Exactly as was the case for $T(1243)$, we check that if a site was active in π, it remains active in π^i unless it falls to the right of position x and to the left of position i, where $n+1$ was inserted.

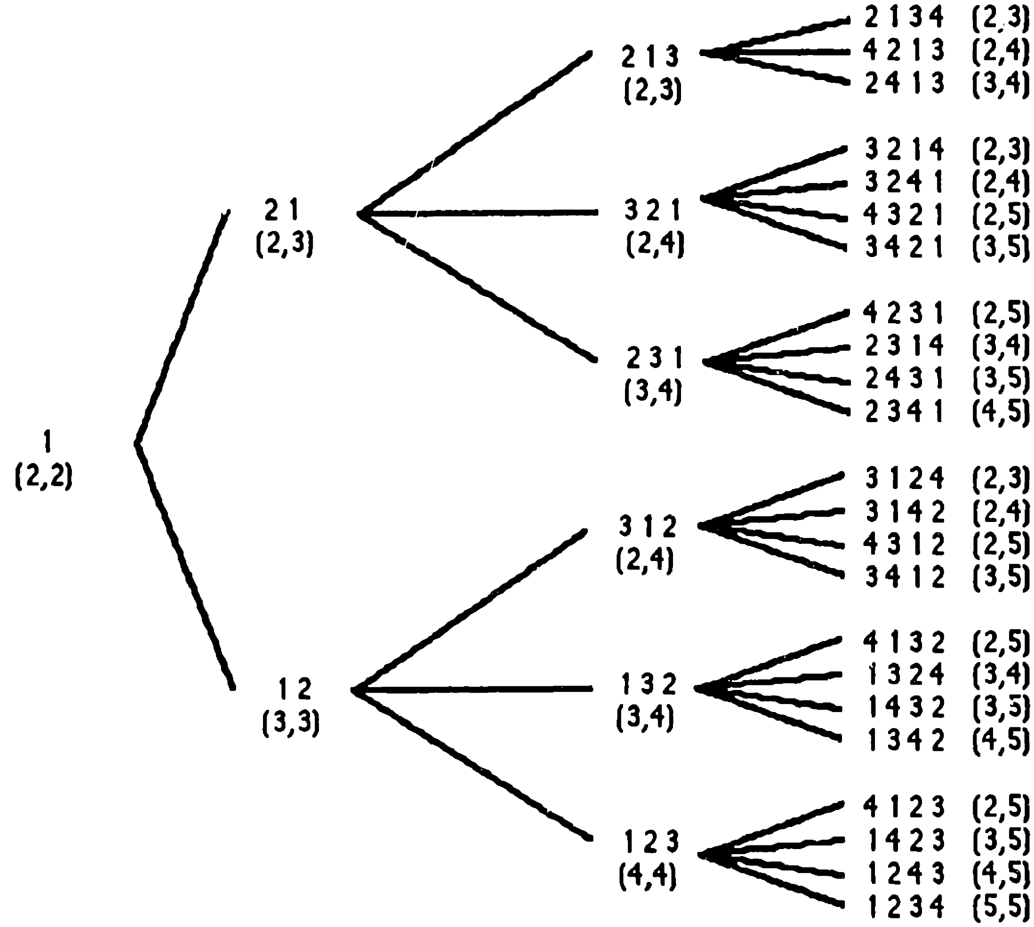
We thus observe that inserting $n+1$ into the ith active site gives rise to a permutation π^i associated in $T(1243)$ with the ordered pair

$$\begin{array}{ll} (2, y+1) & \text{if } i = 1, \\ (i+1, y+1) & \text{if } 2 \leq i \leq x, \\ (x, x+y+1-i) & \text{if } x+1 \leq i \leq y. \end{array}$$

If $i = 1$, $n+1$ is inserted to the left of all elements of the permutation. This removes no active sites, and creates one new one, namely to the left of $n+1$ itself. The first position where a *decrease* is detected is now in the very first position following $n+1$ in the new permutation.

The first x active sites are simply the x leftmost sites. If $2 \leq i \leq x$, then $n+1$ is inserted to the left of the first position in which there is an left-to-right decrease. Since $n+1$ is larger than all other elements of π^i, it is larger than the element to its right, which becomes the new first marker of a left-to-right decrease. Also, all active sites of π remain active sites of π_i, and one new one is introduced, to the left of $n+1$.

Finally, if $i \geq x+1$, then $(p_{x-1}, p_x, n+1)$ is a subsequence of type 213. (Again, note that if $i \geq x+1$, it must be the case that $x \leq n$, so that p_x actually exists.) Then all sites between p_x and $n+1$ become inactive. The sites which remain active are the initial x sites and the final $(y-x+1)-(i-x)$ active sites. □



The tree T(2143) with the labels (x,y) at each node

Combining the results of theorems 3.1.2, 3.1.4, and 3.1.5, we can assert that the three rooted trees $T(1234)$, $T(1243)$ and $T(2143)$ are all isomorphic.

Theorem 3.1.6 $T(1234) \cong T(1243) \cong T(2143)$, *and these isomorphisms are unique.*

Proof: In each case, the permutation at the root, (1), is associated with the ordered pair $(2,2)$. Applied recursively, the structural theorems 3.1.2, 3.1.4, and 3.1.5 ensure the isomorphisms.

Since we can verify from the labels (5) assigned to a set of siblings, that no two siblings anywhere in the tree receive the same ordered pair, the tree has a trivial symmetry group, and the isomorphisms are unique. □

Observe that the isomorphism induce bijections between the sets $S_n(1234)$, $S_n(1243)$ and $S_n(2143)$ for all positive n, these being the nodes on level n of their respective trees. As a consequence we have the following.

Corollary 3.1.7 $|S_n(1234)| = |S_n(1243)| = |S_n(2143)|$, *for all positive* n.

In section 3.5 we will use the observations of this section to create an explicit summation formula for $|S_n(1234)|$. Regev has shown [13] that $|S_n(1234)|$ is asymptotic to $c \cdot \frac{9^n}{n^4}$ for some constant c. (See also section 3.4.)

This result can now be applied to the sets $S_n(1243)$ and $S_n(2143)$ as well. The permutations of $S_n(2143)$ in particular have been extensively studied, as these are precisely the *vexillary permutations.* For an alternative characterization of the vexillary permutations, we quote a lemma from section 2 of a paper by Lascoux and Schützenberger [10].

Let $\pi \in S_n$, and define $\mathcal{I}(\pi)$ as the inversion vector $\mathcal{I}_i = |\{j > i : \pi_j < \pi_i\}|$. Let $\mathcal{P}(\pi)$ be the partition obtained by putting $\mathcal{I}(\pi)$ in nonincreasing order. Then π is vexillary if and only if $\mathcal{P}(\pi)$ is the transpose partition of $\mathcal{P}(\pi^{-1})$.

Example 3.1.8
If $\pi = (3,7,6,1,2,4,5)$, *then* $\mathcal{I}(\pi) = (2,5,4,0,0,0,0)$ *and* $\mathcal{P}(\pi) = (5,4,2)$;
and, $\pi^{-1} = (4,5,1,6,7,3,2)$, *so* $\mathcal{I}(\pi^{-1}) = (3,3,0,2,2,1,0)$ *and* $\mathcal{P}(\pi^{-1}) = (3,3,2,2,1)$.
As the partition $(3,3,2,2,1)$ *is the transpose of* $(5,4,2)$, *the permutations* $(3,7,6,1,2,4,5)$ *and* $(4,5,1,6,7,3,2)$ *are vexillary.*

If $\rho = (3,6,1,7,2,4,5)$, *then* $\mathcal{I}(\pi) = (2,4,0,3,0,0,0)$ *and* $\mathcal{P}(\pi) = (4,3,2)$;
and, $\rho^{-1} = (3,5,1,6,7,2,4)$, *so* $\mathcal{I}(\rho{-}1) = (2,3,0,2,2,0,0)$ *and* $\mathcal{P}(\rho^{-1}) = (3,2,2,2)$.
The transpose of $(4,3,2)$ *is* $(3,3,2,1) \neq (3,2,2,2)$. *Hence,* ρ *and* ρ^{-1} *are not vexillary. This can also be seen from the presence of the forbidden subsequence* 3174 *in* ρ, *and* 3162 *in* ρ^{-1}.

We will establish a second correspondence between the 1234-avoiding permutations and the vexillary permutations. See section 3.3 for an explicit description of this correspondence.

3.1.2 Two trees for each $k > 4$

In the above subsection we saw that the two trees $T(1234)$, $T(1243)$ were isomorphic to one another. We now generalize this result to forbidden subsequences of length k, namely we show that the permutations $\iota_k = (1, 2, 3, \ldots, k-1, k)$ and $\lambda_k = (1, 2, 3, \ldots, k, k-1)$ give rise to isomorphic trees $T(\iota_k)$ and $T(\lambda_k)$.

Since the arguments parallel those of the previous subsection, we present them in somewhat less detail.

Definitions 3.1.9 *For any k, let the permutation ι_k be the all-ascending permutation, $\iota_k(i) = i$ for $1 \leq i \leq k$.*
For any k, let the permutation λ_k be the permutation defined by $\lambda_k(i) = i$ for $1 \leq i \leq k-2$; $\lambda_k(k-1) = k$, $\lambda_k(k) = k-1$.

Now, to any permutation $\pi \in S_n(\iota_k)$, associate the vector $(x_1, x_2, \ldots, x_{k-3}, y)$, where y is the number of active sites of π according to definition 3.1.1, and x_j is the index of the leftmost element of the $j+1$-th basic subsequence, if one exists, and $n+1$ otherwise. That is, $\pi(x_j)$ is the final member of an increasing subsequence of length $j+1$, but not of length $j+2$, and x_j is the smallest index having this property.

In this case, y is the index of the leftmost member of the $k-1$-th basic subsequence, every site to the right of this point being active. Therefore the active sites of π are in fact sites $1, 2, 3, \ldots, y$. One then checks that if π is associated to the vector $(x_1, x_2, \ldots, x_{k-3}, y)$, then π^i is associated to the vector

$$\begin{array}{ll}
(x_1+1, x_2+1, x_3+1, \ldots, y+1) & \text{if } i = 1, \\
(i, x_2+1, x_3+1, \ldots, y+1) & \text{if } 2 \leq i \leq x_1, \\
(x_1, i, x_3+1, \ldots, y+1) & \text{if } x_1 < i \leq x_2, \\
\vdots & \\
(x_1, \ldots, x_r, i, x_{r+1}+1, \ldots, y+1) & \text{if } x_r < i \leq x_{r+1}, \\
\vdots & \\
(x_1, x_2, \ldots, x_{k-3}, i) & \text{if } x_{k-3}i + 1 \leq i \leq y.
\end{array}$$

and $\pi^i \notin S(\iota_k)$ if $i > y$. In particular, the set of vectors associated to the children $\pi^1, \ldots, \pi^y$ of a permutation π depends only on the vector associated to π.

Likewise, to any permutation $\pi \in S_n(\lambda_k)$, associate the vector $(x_1, x_2, \ldots, x_{k-3}, y)$, where y is the number of active sites of π, and x_j is the least index such that $\pi(x_j)$ is the final member of an increasing subsequence of length $j+1$, but not of length $j+2$. If there is no such index, let x_j be $n+1$. Note that all sites to the left of position x_{k-3} are active; the remaining $y - x_{k-3}$ active sites are to the right of this position.

It is again not hard to see that if π is associated to the vector $(x_1, x_2, \ldots, x_{k-3}, y)$, having the y active sites $a(1) < \ldots < a(y)$, then $\pi^{a(i)}$ is associated to the vector

$$\begin{array}{ll}
(x_1+1, x_2+1, x_3+1, \ldots, y+1) & \text{if } i = 1, \\
(i, x_2+1, x_3+1, \ldots, y+1) & \text{if } 2 \leq i \leq x_1, \\
(x_1, i, x_3+1, \ldots, y+1) & \text{if } x_1 < i \leq x_2, \\
\vdots & \\
(x_1, \ldots, x_r, i, x_{r+1}+1, \ldots, y+1) & \text{if } x_r < i \leq x_{r+1}, \\
\vdots & \\
(x_1, x_2, \ldots, x_{k-3}, y + x_{k-3} + 1 - i) & \text{if } x_{k-3}+1 \leq i \leq y.
\end{array}$$

The tables of offset equations above are identical except in the last line. A comparison of the last lines will reveal that if a node in $T(\iota_k)$ and one in $T(\lambda_k)$ are associated with the same vector, then their y children receive the same y vectors. Also, no two children of the same permutation receive the same vector in either tree.

Since the root permutation, (1), is associated with the vector $(2, 2, 2, \ldots, 2, 2)$ in each tree, we conclude that

Theorem 3.1.10 $T(\iota_k) \cong T(\lambda_k)$, *for all* $k \geq 3$.

We have as an immediate consequence,

Corollary 3.1.11 $|S_n(\iota_k)| = |S_n(\lambda_k)|$, *for all* $k \geq 3$.

In the following section we will obtain another bijective proof of this fact, as part of a more general result.

3.2 A generalization of the Simion-Schmidt algorithm

In this section, we generalize the result of section 2.3 to obtain further cases in which $|S_n(\tau)| = |S_n(\rho)|$ for all n.

If π avoids a sequence $\tau = (t_1, \ldots, t_k)$, then there is no subsequence $\pi(i_1), \pi(i_2), \ldots, \pi(i_k)$ of type τ. There may however be numerous subsequences of the same type as some prefix of τ, say $\tau_s = t_1, \ldots, t_s$ for $s < k$.

Definition 3.2.1 *Let $\tau \in S_k$, $\tau = (t_1, \ldots, t_k)$. Also let $\pi \in S_n(\tau)$, $\pi = (p_1, \ldots, p_n)$. For $1 \leq j \leq n$, if r is the largest integer for which there exists a subsequence of the form $p_{i_1}, \ldots, p_{i_r} = p_j$ which is of the same type as $t_1, \ldots, t_r$, then let p_j be a member of the r-th* basic subsequence *of π with respect of τ.*

Thus there are $k-1$ basic subsequences with respect to $\tau \in S_k$ of any $\pi \in S_n(\tau)$, some of them possibly empty. If $\tau = \iota_k = 1, 2, \ldots, k$, definition 3.2.1 coincides with the usual Schensted definition of basic subsequences (section 2.2).

We can establish a general class of bijections using the notion of basic subsequences. The permutations which will be involved in these bijections will each have one of two special forms. Let us say that a permutation τ is of *type one* if $\tau = (t_1, t_2, \ldots, t_{k-2}, k-1, k)$ and of *type two* if $\tau = (t_1, t_2, \ldots, t_{k-2}, k, k-1)$. The statement and proof of the theorem require two further definitions.

Definitions 3.2.2 *If $\tau \in S_k$, let $\tilde{\tau}$ be defined by*

$\tilde{\tau}(j) = \tau(j)$ *for* $j < k-1$

$\tilde{\tau}(k-1) = \tau(k)$

$\tilde{\tau}(k) = \tau(k-1)$

Also let $\bar{\tau} \in S_{k-1}$ be the standard type of the sequence $\tau(1), \tau(2), \ldots, \tau(k-1)$.

The operation of $\tilde{\tau}$ is clearly an involution. We will only be applying this operation to permutations of types one and two. We remark that if τ is of type one, then $\tilde{\tau}$ is of type two, and *vice versa*. Also if τ is of either of type one or of type two, and $\rho = \tilde{\tau}$, then $\bar{\tau} = \bar{\rho}$.

Theorem 3.2.3 *If $\tau \in S_k$ is of types one or two, then $|S_n(\tilde{\tau})| = |S_n(\tau)|$.*

The proof is a generalization of the Simion-Schmidt algorithm presented in 2.3.

Algorithm 3.2.4

Input: *$\pi \in S_n(\tau)$ for $\tau \in S_k$, $\tau(k-1) = k-1, \tau(k) = k$.*

Output: *$\pi^* \in S_n(\tilde{\tau})$.*

Step 1: *Let B be the set of $\pi(j)$ in the $(k-1)$st basic subsequence of π with respect to τ.*

Step 2: *Set $j \leftarrow 0$.*

Step 3: *$j \leftarrow j+1$.*

Step 4: *If $\pi(j)$ is not in the $(k-1)$-th basic subsequence with respect to τ: set $\pi^*(j) = \pi(j)$, go to step 5.*

Otherwise $\pi(j)$ is in the $(k-1)$st basic subsequence. Then let b be the smallest member of B such that there exists $1 \leq i_1 \leq \cdots \leq i_{k-2} < j$ so that $\pi^*(i_1), \ldots, \pi^*(i_{k-2}), b$ is of type $\bar{\tau}$. Let $\pi^*(j) = b$. $B = B \setminus \{b\}$.

Step 5: *If $j = n$, exit; otherwise, go to step 3.*

Lemma 3.2.5 *If $\pi \in S_n(\tau)$ is of type one, let π^* be defined by algorithm 3.2.4. Then $\pi^* \in S_n(\bar{\tau})$.*

Proof: First, we check that the algorithm will terminate having produced some $\pi^* \in S_n$. The only possible difficulty would be if there were at some stage no suitable choice of b remaining in B.

The set B was originally composed of the members of the $(k-1)$st basic subsequence of π with respect to τ. This is a descending subsequence. (If $\pi(j_1)$ and $\pi(j_2)$ are both in the $(k-1)$st basic subsequence and $\pi(j_1) < \pi(j_2)$ with $j_1 < j_2$, then there exists a subsequence $\pi(i_1), \ldots, \pi(i_{k-1}) = \pi(i_j)$ of type $\bar{\tau}$. Then, since $\tau(k-1) = k-1$ and $\tau(k) = k$, it follows that $\pi(j_1)$ is greater than each of $\pi(i_1), \ldots, \pi(i_{k-2})$. As $\pi(j_2)$ is larger still, we see that $\pi(i_1), \ldots, \pi(i_{k-2}), \pi(j_1), \pi(j_2)$ is of type τ. This contradicts $\pi \in S_n(\tau)$, so it follows that the $(k-1)$st basic subsequence is a descending subsequence.)

Suppose we are on pass number j_0 and $\pi(j_0)$ is in the $(k-1)$st basic subsequence, so that we seek a suitable candidate for b. Certainly, the element $\pi(j_0)$ or any larger number would certainly suffice. But if $\pi(j_0)$ is the m-th member from the left of the $(k-1)$st basic subsequence, it is also the mth largest, as it is a descending subsequence. Not all of the m largest members of B can already have been used in the $(m-1)$ positions already encountered which belonged to the $(k-1)$st basic subsequence. So there will always be a suitable b remaining in B.

Now suppose that the permutation generated by the algorithm, π^*, contains a subsequence of type $\bar{\tau}$, say $\pi^*(i_1), \ldots, \pi^*(i_k)$. Then $\pi^*(i_{k-1})$ is the largest member of this subsequence, and $\pi^*(i_k)$ is the next largest.

It cannot be the case that both $\pi^*(i_k)$ and $\pi^*(i_{k-1})$ were both members of B, because the minimality conditions would not have allowed the selection of $\pi^*(i_{k-1})$ as b while the smaller $\pi^*(i_k)$ was still remaining in B, and would have sufficed (since $\pi^*(i_1), \ldots, \pi^*(i_{k-2}), \pi^*(i_k)$ is equally a subsequence of type $\bar{\tau}$).

It remains to show that it is impossible that either of $\pi^*(i_k)$ or $\pi^*(i_{k-1})$ was initially a nonmember of B. To do this, we will consider two cases. First note that if either of $\pi^*(i_k)$ or $\pi^*(i_{k-1})$ was not a member of B, then it was fixed by the algorithm, $\pi^*(i_k) = \pi(i_k)$ [or $\pi^*(i_{k-1}) = \pi(i_{k-1})$].

For the first case, suppose each of $\pi^*(i_1), \ldots, \pi^*(i_{k-2})$ was also fixed. In this case the subsequence $\pi(i_1), \ldots, \pi(i_{k-2}), \pi(i_k)$ is of type $\bar{\tau}$, meaning that $\pi(i_k)$ is in the $k-1$st basic subsequence of π, in other words that it began in B. (The proof for $\pi^*(i_{k-1})$ is identical.)

For the remaining case, suppose one of $\pi^*(i_1), \ldots, \pi^*(i_{k-2})$ was also fixed. Suppose it is $\pi^*(i_s)$. Then there exists a subsequence of type $\bar{\tau}$ ending with $\pi^*(i_s)$, and with all its members $\leq \pi^*(i_s) < \pi^*(i_k)$. Each member of this subsequence was fixed or one was not, and by continuing we must reach a subsequence of type $\bar{\tau}$ which was entirely fixed, except for possibly the final element, and which lies entirely to the left of $\pi(i_k)$ [or $\pi(i_{k-1})$]. Each of its elements is by construction smaller than $\pi(i_k)$ [or $\pi(i_{k-1})$], so replace the final element by $\pi(i_k)$ [or $\pi(i_{k-1})$] to obtain an entirely fixed subsequence of type $\bar{\tau}$. The existence of this subsequence in π would require that $\pi(i_k)$ belong to the $(k-1)$st basic subsequence, contradicting our supposition.

This completes the proof that π^* avoids $\tilde{\tau}$. □

The following algorithm can be seen to be the inverse of algorithm 3.2.4, thus establishing the bijection of 3.2.3.

Algorithm 3.2.6

Input: $\pi \in S_n(\tau)$ *for* $\tau \in S_k$, $\tau(k-1) = k, \tau(k) = k-1$.

Output: $\pi^* \in S_n(\tilde{\tau})$.

Step 1: *Let B be the set of $\pi(j)$ in the $(k-1)$st basic subsequence of π with respect to τ.*

Step 2: *Set $j \leftarrow 0$.*

Step 3: $j \leftarrow j + 1$.

Step 4: *If $\pi(j)$ is not in the $(k-1)$-st basic subsequence with respect to τ, set $\pi^*(j) = \pi(j)$. Go to step 5.*
Otherwise $\pi(j)$ is in the $(k-1)$st basic subsequence. Then let b be the largest member of B. Let $\pi^*(j) = b$. $B = B \setminus \{b\}$.

Step 5: *If $j = n$, exit; otherwise, go to step 3.*

It is easy to see that in the case $k = 3$ algorithms 3.2.4 and 3.2.6 reduce to the Simion-Schmidt algorithms of 2.3. As in that case, the algorithms produce bijections from one set of restricted permutations, $S_n(\tau)$, to another, $S_n(\tilde{\tau})$, while fixing the intersection of the two sets.

3.3 Tables for $k = 4, 5, 6$

The tables given on the following pages summarize the present situation for forbidden subsequences of length $k = 4, 5, 6$. In these tables, values of $|S_n(\tau)|$ are given for small n and all τ of length ≤ 6, In the tables for 4 and 5, permutations which are related to one another by the standard operations of conjugation, reversal and inversion are grouped together in boxes. For $k = 6$, we have chosen only one exemplar of each class, namely the lexicographically smallest.

Permutations which have $\tau(k-1) = k$ and $\tau(k) = k - 1$ are marked with asterisks, and permutations which are related by the application of theorem 3.2.3 have been placed on the same level.

As an instance of the interplay between the standard operations and the new considerations of theorem 3.2.3, note that we have bijections between $S_n(1234)$ and $S_n(1243)$ by 3.2.3, between $S_n(1243)$ and $S_n(3421)$ by reversal, between $S_n(3421)$ and $S_n(2134)$ by complementations, and between $S_n(2134)$ and $S_n(2143)$ by 3.2.3. Composing these bijections, we have a second bijection between $S_n(1234)$, the permutations with no ascending subsequence of length 4, and $S_n(2143)$, the vexillary permutations. This bijection is completely distinct from the one given in section 3.1, where the vexillary permutations were defined preceding example 3.1.8.

As can be seen from the proliferation of values at the right of the tables, nearly all cases in which equality of $S_n(\tau)$ and $S_n(\sigma)$ holds for all n have been established. There are, however, a few outstanding cases. The most obvious of these are the large number of τ for which the number of τ-avoiding permutations equals the number of permutations which avoid ι_k, the identity. One conjecture would tie together nearly all of these.

Conjecture 3.3.1 $|S_n(1,2,...,k)| = |S_n((r+1),(r+2),\ldots,k,1,2,\ldots,r)|$ *for all natural numbers* n, k, r.

Those permutations which would be affected by this conjecture are grouped together by double bars in the tables.

It also remains possible that equality obtains for the first two groups in the table for $k = 4$. Let us add this as a conjecture.

Conjecture 3.3.2 $|S_n(2,4,3,1)| = |S_n(2,4,1,3)|$ *for all positive* n.

One approach to conjecture 3.3.2 would be to show an isomorphism between the trees $T(4132)$ and $T(3142)$. These trees are in fact identical down to depth six.

Finally, there are several groups in the table for $k = 6$ for which equality obtains through $n = 6$, but which are not covered by any of the previous cases. These have been marked with special symbols in the table. Close inspection of these equivalence classes or groups of classes shows that each pair is represented by permutations having a similar form. In particular, all the six permutations of the form $abc456$ are matched to the six corresponding permutations $abc654$ by these marks.

The first pair, marked by diamonds, contains the permutations 231456 and 231654, as well as their inverses 312456 and 312654.

The second pair, marked by hearts, contains 123456 and 123654. The permutations 213456 and 213654 are related to these by the result of 3.2.3, and so can be found in these boxes as well.

The third pair, marked by clubs, contains 321456 and 321654. Evidently, a bijection linking this pair would also establish equality for the pair marked by hearts, if conjecture 3.3.1 had also been established.

The final pair, marked by spades, contains 132456 and 132654.

We collect these four cases into one conjecture.

Conjecture 3.3.3 $|S_n(a,b,c,4,5,6)| = |S_n(a,b,c,6,5,4)|$ *for all positive* n.

The establishment of the above conjectures would take care of all the candidates for bijections involving forbidden subsequences of length less than or equal to 6.

Conjecture 3.3.3 amounts to a generalization of theorem 3.2.3. An even more general form is evident, which we can state even though we have no further data:

Conjecture 3.3.4

$$|S_n(a_1, a_2, \ldots, a_{k-r}, k-r+1, k-r+2, \ldots, k-1, k)| = |S_n(a_1, a_2, \ldots, a_{k-r}, k, k-1, \ldots, k-r+2, k$$

for all positive n, r, k.

When $r = 2$, conjecture 3.3.4 reduces to theorem 3.2.3; a natural way to attack this conjecture would be to seek to extend the proof of 3.2.3 by fixing all but the last $r-1$ basic subsequences.

Also observe that conjecture 3.3.4 is stronger than conjecture 3.3.1, as 3.3.1 follows as a consequence of 3.3.4, lemma 1.2.2 on reversals and lemma 1.2.3 on complements. A more direct proof of conjecture 3.3.1 would seemingly require an entirely new construction technique, perhaps again involving basic subsequences.

One approach to conjecture 3.3.2 would be to show an isomorphism between the trees $T(4132)$ and $T(3142)$. These trees are in fact identical down to depth six.

<table>
<tr><th colspan="3">τ</th><th>$S_5(\tau)$</th><th>$S_6(\tau)$</th><th>$S_7(\tau)$</th><th>$S_8(\tau)$</th><th>$S_9(\tau)$</th><th>$S_{10}(\tau)$</th></tr>
<tr><td colspan="3">1342
1423
2314
2431
3124
3241
4132
4213</td><td>103</td><td>512</td><td>2740</td><td>15 485</td><td>91 245</td><td>555 662</td></tr>
<tr><td colspan="3">2413
3142</td><td>103</td><td>512</td><td>2740</td><td>15 485</td><td>91 245</td><td>555 662</td></tr>
<tr><td>1234*
4321</td><td>1243*
2134*
3421
4312</td><td>2143*
3412</td><td rowspan="2">103</td><td rowspan="2">513</td><td rowspan="2">2761</td><td rowspan="2">15 767</td><td rowspan="2">94 359</td><td rowspan="2">586 590</td></tr>
<tr><td colspan="3">1423
2341
3214
4123</td></tr>
<tr><td colspan="3">1324
4231</td><td>103</td><td>513</td><td>2762</td><td>15 793</td><td>94 776</td><td>591 950</td></tr>
</table>

τ				$S_6(\tau)$	$S_7(\tau)$	$S_8(\tau)$	$S_9(\tau)$	$S_{10}(\tau)$
41352 25314				694	4578	33 184	258 757	2 136 978
31524 35142 42513 24153				694	4579	33 216	259 401	2 147 525
23514 25134 41532 43152	41253 31452 25413 35214			694	4579	33 218	259 483	2 149 558
24513 31542 35124 42153	41523 25143 34152 32514			694	4580	33 249	260 092	2 159 381
13542 24531 42135 53124	15243 51423 32415 34251			694	4580	33 252	260 202	2 161 837
14352 25341 41325 52314	15324 51342 24315 42351			694	4580	33 252	260 204	2 161 930
13524 24135 42531 53142	14253 31425 52413 35241			694	4580	33 254	260 285	2 163 930
13425 14235 52431 53241				694	4580	33 256	260 370	2 166 120
13452 25431 41235 53214	15234 51432 23415 43215			694	4581	33 283	260 805	2 171 393
14523 32541 34125 52143				694	4581	33 284	260 847	2 172 454
14532 23541 43125 52134	15423 51243 34215 32451			694	4581	33 285	260 886	2 173 374
12453 31245■ 35421 54213	12534 23145■ 54132 43521	21453 31254■ 35412 45213	21534 23154■ 45132 43512	694	4581	33 286	260 927	2 174 398
15342 24351 42315 51324				694	4581	33 287	260 967	2 175 379
12345■ 54321	45312 21354■	12354■ 45321	21345■ 54312	694	4582	33 324	261 808	2 190 688
15432 23451 43215 51234				694	4582	33 324	261 808	2 190 688
21543 32154■ 34512 45123		12543 32145■ 34521 54123		694	4582	33 324	261 808	2 190 688
12435 13245■ 53421 54231		13254■ 21435 45231 53412		694	4582	33 325	261 853	2 191 902

r	$S_7(r)$	$S_8(r)$	$S_9(r)$	$S_{10}(r)$
246153	5003	39 424	344 571	3 282 986
246315	5003	39 424	344 572	3 283 043
253614	5003	39 424	344 574	3 283 168
251463	5003	39 424	344 575	3 283 227
251364	5003	39 424	344 579	3 283 483
136425	5003	39 424	344 580	3 283 521
254163	5003	39 425	344 611	3 283 955
263415	5003	39 425	344 614	3 284 117
256314	5003	39 425	344 616	3 284 248
251634	5003	39 425	344 619	3 284 418
351624	5003	39 425	344 620	3 284 441
236415	5003	39 425	344 620	3 284 457
251634	5003	39 425	344 622	3 284 567
235164	5003	39 425	344 622	3 284 568
246135	5003	39 425	344 622	3 284 571
241635	5003	39 425	344 622	3 284 575
231653	5003	39 425	344 624	3 284 706
146253	5003	39 425	344 627	3 284 887
153624	5003	39 425	344 628	3 284 941
136524	5003	39 425	344 629	3 284 991
134625	5003	39 425	344 629	3 285 006
146325	5003	39 425	344 630	3 285 060
135264	5003	39 425	344 631	3 285 115
136245	5003	39 425	344 633	3 285 226
236145	5003	39 426	344 661	3 285 505
236514	5003	39 426	344 662	3 285 559
245163	5003	39 426	344 666	3 285 746
256143	5003	39 426	344 667	3 285 813
235614	5003	39 426	344 668	3 285 874
256413	5003	39 426	344 668	3 285 891
254613	5003	39 426	344 669	3 285 926
256134	5003	39 426	344 669	3 285 936
236154	5003	39 426	344 671	3 286 054
246513	5003	39 426	344 672	3 286 086
243615	5003	39 426	344 672	3 286 113
145263	5003	39 426	344 672	3 286 128
153462	5003	39 426	344 674	3 286 231
234615	5003	39 426	344 675	3 286 273
135624	5003	39 426	344 675	3 286 293
145362	5003	39 426	344 676	3 286 337
135642	5003	39 426	344 678	3 286 444
136452	5003	39 426	344 678	3 286 445
143625	5003	39 426	344 678	3 286 463
164352	5003	39 426	344 679	3 286 521
153642	5003	39 426	344 680	3 286 576
146352	5003	39 426	344 680	3 286 586
136254	5003	39 426	344 681	3 286 660
163542	5003	39 426	344 682	3 286 686
135426	5003	39 426	344 682	3 286 719
135246	5003	39 426	344 686	3 286 953
326154	5003	39 427	344 724	3 287 748
154623	5003	39 427	344 724	3 287 749
146523	5003	39 427	344 724	3 287 751
143652	5003	39 427	344 725	3 287 819
134526	5003	39 427	344 725	3 287 840
134652	5003	39 427	344 726	3 287 851
143562	5003	39 427	344 726	3 287 863
146532	5003	39 427	344 726	3 287 877
245613	5003	39 427	344 727	3 287 904
154362	5003	39 427	344 727	3 287 939
135462	5003	39 427	344 727	3 287 971
145236	5003	39 427	344 728	3 287 974
154632	5003	39 427	344 728	3 287 993
136542	5003	39 427	344 728	3 287 995
145326	5003	39 427	344 729	3 288 076
125463 215463	5003	39 427	344 731	3 288 165
124653 214653	5003	39 427	344 731	3 288 170
153426	5003	39 427	344 732	3 288 259
124635 214635	5003	39 427	344 733	3 288 282
124536 134265	5003	39 427	344 735	3 288 426
231564	5003	39 428	344 772	3 289 163
231645	5003	39 428	344 772	3 289 163
156423	5003	39 428	344 776	3 289 398
163452	5003	39 428	344 777	3 289 452
124563 214563	5003	39 428	344 777	3 289 454
145632	5003	39 428	344 778	3 289 499
145623	5003	39 428	344 778	3 289 500
125634 215634	5003	39 428	344 778	3 289 522
125643 215643	5003	39 428	344 779	3 289 579
134562	5003	39 428	344 781	3 289 663
156342	5003	39 428	344 781	3 289 688
126453 216453	5003	39 428	344 781	3 289 705
164532	5003	39 428	344 782	3 289 753
156432	5003	39 428	344 783	3 289 800
123564 ♢ 213564	5003	39 428	344 784	3 289 863
231654 ♢	5003	39 428	344 784	3 289 863
132564	5003	39 428	344 784	3 289 875
123456 ♡ 123465 213465	5003	39 429	344 837	3 291 590
123654 ♡ 213654 ♣	5003	39 429	344 837	3 291 590
126543 216543	5003	39 429	344 837	3 291 590
165432	5003	39 429	344 837	3 291 590
321654 ♣	5003	39 429	344 837	3 291 590
123546 ♠ 132465	5003	39 429	344 838	3 291 652
132654 ♠	5003	39 429	344 838	3 291 652
124356 124365 214365	5003	39 429	344 838	3 291 656
154326	5003	39 429	344 838	3 291 662
125436 143265	5003	39 429	344 838	3 291 666
132546	5003	39 429	344 839	3 291 715

3.4 Future directions

Beyond the two conjectures of the previous section, future directions are not as easy to see. On the one hand, it would obviously be nice to be able to enumerate $S_n(\tau)$ for every n and every τ, but the discouragingly chaotic proliferation of values in the above tables suggests that this will not be possible. Ira Gessel asks, in the final paragraph of [6], whether each of these functions of n is *P-recursive*, meaning that the function satisfies a linear homogeneous recurrence with polynomial coefficients.

We pose a further series of questions, the settling of which would constitute an acceptable assault on the problem of forbidden subsequences.

Question 3.4.1 *When is it true that $|S_n(\tau)| = |S_n(\rho)|$ for all n? In particular, are there any cases other than those which can be established by repeated application of lemmas 1.2.2, 1.2.3 and 1.2.4, theorem 3.2.3 and conjectures 3.3.4 and 3.3.2?*

Question 3.4.2 *If $|S_N(\tau)| < |S_N(\rho)|$ for some N, will it be true that $|S_n(\tau)| < |S_n(\rho)|$ for all $n > N$? Given two permuations τ and ρ of the same length, how can we decide which of the sets $S_n(\tau)$ and $S_n(\rho)$ will be the larger?*

Question 3.4.3 *Is it true that for any k and for any $\tau, \rho \in S_k$, the values of $|S_n(\tau)|$ are asymptotically equal as n increases?*

If the answer to question 3.4.3 is affirmative, then the work of Regev [13] provides a nice asymptotic formula. For every $k \geq 2$, Regev provides an asymptotic analysis of the values $|S_n(\iota_k)|$, where ι_k is the ascending permutation of length k. Specifically, he shows that

$$|S_n(\iota_k)| \sim \mathbf{c}_k \cdot \frac{(k-1)^{2n}}{n^{k^2-2k/2}}$$

where $\mathbf{c}_k$ is an explicitly determined constant involving a multiple integral. Regev's analysis exploits the Robinson-Schensted correspondence. A permutation of $S_n(\iota_k)$ has at most $k-1$ basic subsequences and thus no more than $k-1$ columns in the associated P and Q tableaux. The number of such permutations is thus the square of the number of tableaux of a given shape, summed over all shaps with no more than $k-1$ columns. Regev determines those shapes of tableaux which dominate the sum as $n \to \infty$.

It seems that the most promising proof technique for the three above questions, especially for 3.4.2, would be to generalize the tree construction techniques of sections 2.5.1

and 3.1. At this point, there is no example of a pair of permutations of the same length for which it is known that $|S_n(\tau)| < |S_n(\rho)|$ for all n sufficiently large. It would be good to find an injection proving such a result, and one approach might be to consider the stronger case that the tree associated to τ be a subtree of the tree associated to ρ.

In particular, we propose the following pairs of trees as candidates.

Question 3.4.4 *Is it true that $T(1342)$ is a proper subtree of $T(1432)$?*
Is it true that $T(1423)$ is a proper subtree of $T(1324)$?

Note that none of these four trees has had its structure properly elucidated.

3.5 A recurrence for $k = 4$

The observations about the structure of the trees of section 3.1.1 lead directly to recurrence formulæ, which make it relatively easy to calculate the exact size of $S_n(1234)$. As remarked at the end of section 3.1, the asymptotic analysis of this case has already been carried out by Regev [13]. Gessel [6] also finds an explicit formula for $|S_n(1234)|$, involving a summation over a product of binomial coëfficients and a rational function.

The recurrence derived from the trees in chapter 3 remains a reasonable efficient way of calculating the sizes of $S_n(1234)$. The coëfficients arising in this recurrence are of independent interest; we discuss them here.

Definition 3.5.1 *Let $P(n;r,s)$ denote the number of permutations of $1,2,\ldots,n$ avoiding the pattern 1234 with length of initial decreasing sequence equal to $r-1$ (or, with first increase in position r) and with s active sites.*

Corresponding to each n, then, we have a matrix of order $n+1$, which we can call $P(n)$, with the coefficient $P(n;r,s)$ counting the number of permutations on level n of the tree $T(1234)$ having the label (r,s). Theorem 3.1.2 associates to each child of a permutation a different ordered pair in a prescribed manner. The set of labels of the children depend only on the label of the permutation. The $P(n;r,s)$ permutations on level n having the label (r,s) therefore each contributes one to the value of certain coefficients in the matrix $P(n+1)$.

Therefore, in computing the coefficients of $P(n+1)$ we increment each of the coefficients

$$\begin{aligned}P(n+1;2,y+1), P(n+1;3,y+1),\ldots,P(n+1;x,y+1), P(n+1;x,x+1),\\ P(n+1;x,x+2),\ldots,P(n+1;x,y), P(n+1;x+1,y+1)\end{aligned} \tag{6}$$

by the value $P(n;x,y)$ as a result of the coefficient $P(n;x,y)$ in the previous matrix.

Collecting together the contributions made to each position in $P(n+1)$, we have the following recursive formula.

Theorem 3.5.2 *The coefficient $P(n+1;r,s)$ is equal to*

$$
\begin{array}{lll}
0 & \text{if } r = 1, & (7)\\
0 & \text{if } s \le r < n+1, & (8)\\
1 & \text{if } s = r = n+1, & (9)\\
\sum_{i=r-1}^{s-2} P(n;i,s-1) + \sum_{j=s}^{n+1} P(n;r,j) & \text{if } s > r. & (10)
\end{array}
$$

Proof: The summation formula (10) for the entries above the main diagonal can be checked by examining (6) to see exactly which entries of $P(n)$ contribute to $P(n+1;r,s)$. That is, for which (x,y) does (r,s) have one of the forms listed in (6)? As this is a straightforward exercise, we dispense with formalities, and include the following illustration which may be illustrative. A box has been drawn around entry in position $(5,6)$ of $P(6)$. The entries of $P(6)$ which contribute to $P(7;5,6)$ are shaded. Actually, we have truncated the first sum in expression (10) at the last nonzero term, while in the diagram we have included the zero terms on and below the main diagonal, in effect extending the sum to $n+1$. Summing the entries in the shaded shape we obtain $P(7;5,6) = 50$.

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 47 & 47 & 47 & 42 & 42\\
0 & 0 & 0 & 47 & 47 & 42 & 42\\
0 & 0 & 0 & 0 & 26 & 26 & 28\\
0 & 0 & 0 & 0 & 0 & \boxed{10} & 14\\
0 & 0 & 0 & 0 & 0 & 0 & 5\\
0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
$$

None of the forms listed in (6) has $r = 1$, so it is especially easy to see that $P(n+1;0,s) = 0$.

Also, only the form $(x+1, y+1)$ has the first index greater than or equal to the second index, so (8) and (9) follow readily.

The value 1 given for the coefficient $P(n+1; n+1, n+1)$ in (9) is actually just a special case of (10). $\square$

Note that the coefficients $P(n; r, s)$ have alternative interpretations in terms of the trees $T(1243)$ and $T(2143)$. Thus, $P(n; r, s)$ is the number of 1243-avoiding permutations of $1, 2, 3, \ldots, n$ with initial decreasing sequence of length $r-1$ and with s active sites. Also, $P(n; r, s)$ is the number of 2143-avoiding permutations with initial increasing sequence of length $r-1$ and with s active sites. Other interpretations can be obtained by considering the action of the operations of reversal, complementation, and inversion on permutations.

The matrices $P(n)$ for $n \leq 8$ are appended. By summing over all entries in $P(n)$, we can determine the number of 1234-avoiding permutations in an efficient manner. These totals are appended, and can be compared with those given in the table for $n = 4$ in section 3.3. In addition, we list the row and column sums for each matrix. These have natural interpretations; let us give them names.

Definition 3.5.3 $R(n; r) = \sum_{j=1}^{n+1} P(n; r, j)$

Definition 3.5.4 $C(n; s) = \sum_{i=1}^{n+1} P(n; i, s)$

$R(n; r)$, the rth row sum of matrix $P(n)$, is the number of permutations of $1, 2, 3, \ldots, n$ which avoid the pattern 1234 and have initial decreasing sequence of length $r-1$. $C(n; s)$, the sth column sum, is the number of permutations of length n which avoid 1234 and have s active sites.

It would be nice to have asymptotic formulas, depending on n, r and s, for the individual matrix entries or for the row and column sums. We do not know of any such formulas in a simple form.

We are able, however, to express all the entries in the last column in a closed form.

Theorem 3.5.5 *The number of permutations of* $1, 2, 3, \ldots, n$ *having no pattern* 1234, *an initial decreasing sequence of length* $i-1$ *and all sites active, namely* $P(n; i, n+1)$, *is*

$$\binom{2n-i}{n-i+1} - \binom{2n-i}{n-i} \tag{11}$$

Proof: The coefficients in the righthand column of $P(n)$ depend only on the coefficients in the righthand column of $P(n-1)$. That is to say, the second sum in (10) can be seen to be empty. So in this case, we have a particularly simple recurrence for the numbers $P(n; r, n+1)$, namely

$$P(n; r, n+1) = \sum_{i=r-1}^{n} P(n-1; i, n) \tag{12}$$

where n is a positive integer and $2 \leq r \leq n+1$.

We then check that the numbers (11) obey the same recurrence.

First note the agreement of the two for $n=2$ and $r=2$:

$$\binom{2n-r}{n-r+1} - \binom{2n-r}{n-r} = \binom{2}{1} - \binom{2}{0} = 2-1 = 1 = P(2;2,3)$$

and for $n=2$, $r=3$:

$$\binom{2n-r}{n-r+1} - \binom{2n-r}{n-r} = \binom{2}{0} - \binom{2}{-1} = 1-0 = 1 = P(2;3,3)$$

Then observe that, for $n>2$ and $2 \leq r \leq n+1$,

$$\begin{aligned}
\sum_{i=r-1}^{n}\left(\binom{2n-2-i}{n-i} - \binom{2n-2-i}{n-1-i}\right) &= \sum_{i=r-1}^{n}\binom{2n-2-i}{n-i} - \sum_{i=r-1}^{n}\binom{2n-2-i}{n-1-i} \\
&= \sum_{i=0}^{n-r+1}\binom{n-2+i}{i} - \sum_{i=0}^{n-r+1}\binom{r-2+i}{i-1} \\
&= \sum_{i=0}^{n-r+1}\binom{n-2+i}{i} - \sum_{i=0}^{n-r}\binom{n-1+i}{i} \\
&= \binom{2n-r}{n-r+1} - \binom{2n-r}{n-r}
\end{aligned}$$

The last line follows from the application to each sum of the binomial coefficient identity

$$\sum_{k=0}^{b}\binom{a+k}{k} = \binom{a+b+1}{b}$$

which itself follows from the standard Pascal triangle recurrence by telescoping.

Since the terms of (11) obey the desired recurrence, they give the coefficients in the rightmost column of the matrices $P(n)$. □

Setting $i=2$ in the above theorem gives us a familiar formula.

Corollary 3.5.6 *The number of permutations of* $1,2,3,\ldots,n$ *having no pattern* 1234, *an initial decreasing sequence of length* 1 *and all sites active, namely* $P(n;2,n+1)$, *is*

$$\binom{2n-2}{n-1} - \binom{2n-2}{n-2}$$

We recognize this to be the Catalan number c_{n-1}. We also have another interpretation for the coefficient $P(n;2,n+1)$, as substituting $r = 2$ and $s = n+1$ into (10) shows.

$$P(n;2,n+1) = \sum_{i=1}^{n} P(n-1;i,n) = C(n-1;n)$$

Thus the final column sum of matrix $P(n-1)$ is also c_{n-1}. But this is no surprise! The entries of the final column count the 1234-avoiding permutations with all sites active. But these are just the 123-avoiding permutations. Similar interpretations for the coefficents $P(n;r,s)$ in terms of 1243-avoiding permutations, and so on, reduce to similarly familiar results when we consider the final column sum and the top right entry. But the more general results of theorem 3.5.5 are new to us.

0	1	
0	0	0
0	1	1

$n = 1$ $\sum = 1$

0	0	2	
0	0	0	0
0	0	1	1
0	0	1	1

$n = 2$ $\sum = 2$

0	0	1	5	
0	0	0	0	0
0	0	1	2	3
0	0	0	2	2
0	0	0	1	1

$n = 3$ $\sum_{rs} = 6$

0	0	3	6	14	
0	0	0	0	0	0
0	0	3	3	5	11
0	0	0	3	5	8
0	0	0	0	3	3
0	0	0	0	1	1

$n = 4$ $\sum_{rs} = 23$

0	0	11	22	28	42	
0	0	0	0	0	0	0
0	0	11	11	11	14	47
0	0	0	11	11	14	36
0	0	0	0	6	9	15
0	0	0	0	0	4	4
0	0	0	0	0	1	1

$n = 5$ $\sum_{rs} P(n;r,s) = 103$

0	0	47	94	120	120	132	
0	0	0	0	0	0	0	0
0	0	47	47	47	42	42	225
0	0	0	47	47	42	42	178
0	0	0	0	26	26	28	80
0	0	0	0	0	10	14	24
0	0	0	0	0	0	5	5
0	0	0	0	0	0	1	1

$n = 6$ $\sum_{rs} P(n;r,s) = 513$

0	0	225	450	577	585	495	429	
0	0	0	0	0	0	0	0	0
0	0	225	225	225	204	162	132	1173
0	0	0	225	225	204	162	132	948
0	0	0	0	127	127	106	90	450
0	0	0	0	0	50	50	48	148
0	0	0	0	0	0	15	20	35
0	0	0	0	0	0	0	6	6
0	0	0	0	0	0	0	1	1

$n = 7$ $\qquad \sum_{rs} P(n; r, s) = 2761$

0	0	1173	2346	3021	3100	2695	2002	1430	
0	0	0	0	0	0	0	0	0	0
0	0	1173	1173	1173	1075	879	627	429	6529
0	0	0	1173	1173	1075	879	627	429	5356
0	0	0	0	675	675	577	423	297	2647
0	0	0	0	0	275	275	219	165	934
0	0	0	0	0	0	85	85	75	245
0	0	0	0	0	0	0	21	27	48
0	0	0	0	0	0	0	0	7	7
0	0	0	0	0	0	0	0	1	1

$n = 8$ $\qquad \sum_{rs} P(n; r, s) = 15767$

In addition to the coefficients in the rightmost column, explicitly determined in theorem 3.5.5, we can determine the single coefficient $P(n; n-1, n)$. The result is trivial, but the technique is illustrative.

Proposition 3.5.7 *The number of permutations of length n with no subsequence of type 1234, initial decreasing subsequence of length $n-2$ and all sites active except one, namely $P(n; n-1, n)$, is* $\binom{n-1}{2}$.

Proof: Let us abbreviate the desired one-parameter variable $P(n; n-1, n)$ by a_n. Then taking the recurrence from theorem 3.5.2 and substituting the appropriate value from 3.5.5 shows that a_n satisfies a first-order linear recurrence.

$$\begin{aligned} a_{n+1} &= a_n + \binom{n}{1} - \binom{n}{0} \\ &= a_n + (n-1) \end{aligned} \tag{13}$$

The general solution to this recurrence is $a_n = a_1 + \sum_{k=1}^{n-1} k - 1$. Since $a_1 = 0$, a_n is just the $n-1$th triangular number, $\binom{n-1}{2}$. □

We can achieve the same result combinatorially. A permutation with initial decreasing sequence of length $n-2$ has its first increase in position $n-1$. If it is a 1234-avoiding permutation, with one inactive site, then the inactive site is the last one, and element in position n is larger than the one in position $n-1$. But *any* permutation π having $\pi(1) > \pi(2) > \cdots > \pi(n-2) < \pi(n-1) < \pi(n)$ will be 1234-avoiding. To count these, we need only assign $\pi(n-2)$, the smallest element, to be 1, and select two elements to be $\pi(n-1)$ and $\pi(n)$. There are $\binom{n-1}{2}$ ways to select two elements from the set 2,3,. . . ,n.

The approach used in the proof of proposition 3.5.7 can also be exploited to give successive terms of the rest of the second column from the right, working from the bottom up. For a taste, we calculate one more term.

Proposition 3.5.8 $P(n; n-2, n) = \binom{n}{3} + \binom{n-1}{3} - (n-2)$

Proof: Let b_n stand in for $P(n; n-2, n)$. We check that b_n satisfies the following recurrence. Here the first driving term comes from the rightmost column; the second comes from the bottom element in the second column, which we have just determined.

$$b_{n+1} = b_n + (\binom{n+1}{2} - \binom{n+1}{1}) + \binom{n-1}{2}$$

We start with the intial value $b_3 = 0$, and obtain the following solution for $n \geq 4$.

$$\begin{aligned} b_n &= \sum_{k=4}^{n} \binom{k}{2} - \sum_{k=4}^{n} \binom{k}{1} + \sum_{k=2}^{n-2} \binom{k}{2} \\ &= \left(\sum_{k=2}^{n} (\binom{k}{2}) - 4\right) - \left(\sum_{k=1}^{n} (\binom{k}{1}) - 6\right) + \sum_{k=2}^{n-2} \binom{k}{2} \end{aligned}$$

$$\begin{aligned}
&= \left(\sum_{k=0}^{n-2}\binom{k+2}{2} - 4\right) - \left(\sum_{k=0}^{n-1}\binom{k+1}{1} - 6\right) + \sum_{k=0}^{n-4}\binom{k+2}{2} \\
&= \binom{n+1}{3} - \binom{n+1}{2} - 4 + 6 + \binom{n-1}{3} \\
&= \left[\binom{n}{3} + \binom{n}{2}\right] - \left[\binom{n}{2} + \binom{n}{1}\right] + 2 + \binom{n-1}{3} \\
&= \binom{n}{3} - n + 2 + \binom{n-1}{3}
\end{aligned}$$

□

This procedure can be repeated to determine all the entries of the nth column. But it turns out that the nth column sum, $C(n;n)$, is considerably easier to determine than the individual coëfficients of the nth column. We give a combinatorial proof.

Theorem 3.5.9 $C(n;n) = \binom{2(n-1)}{n-3}$

Proof: We wish to count the number of 1234-avoiding permutations which have all sites active except for the rightmost. We count instead the 1234-avoiding permutations which have the first n sites active, and which may or may not have an active rightmost site. These are obtained by first choosing any element to be the rightmost element, and then arranging the remaining elements into a 1234-avoiding permutation of length $n-1$ with all sites active.

$$n \cdot c_{n-1} = C(n;n) + C(n;n+1)$$

Here the term on the left comes from choosing a rightmost element in one of n ways, and choosing a 123-avoiding permutation of length $n-1$ in one of c_{n-1} ways. The first term on the right is the number we seek, and the second term on the right is the number of permutations with all sites active.

$$\begin{aligned}
n \cdot c_{n-1} &= C(n;n) + C(n;n+1) \\
n \cdot \frac{1}{n}\binom{2n-2}{n-1} &= C(n;n) + \frac{1}{n+1}\binom{2n}{n} \\
C(n;n) &= \frac{(2n-2)!}{(n-1)!(n-1)!} - \frac{2n!}{(n+1)!(n)!} \\
&= \frac{(n+1)(n)(n)(2n-2)! - (2n)(2n-1)(2n-2)!}{(n+1)!n!}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{[(n^3+n^2)-(4n^2-2n)]\cdot(2n-2)!}{(n+1)!n!} \\
&= \frac{[n^3-3n^2+2n]\cdot(2n-2)!}{(n+1)!n!} \\
&= \frac{(n)(n-1)(n-2)(2n-2)!}{(n+1)!n!} \\
&= \frac{(2n-2)!}{(n+1)!(n-3)!} \\
&= \binom{2n-2}{n-3}
\end{aligned}
$$

□

Inspection of the matrices $P(n)$ reveals that the first two nonzero rows have all entries equal, as do the lowest two nonzero diagonals. These are both straightforward consequences of theorem 3.5.2, especially in regard to the disposition of zero coefficients.

Proposition 3.5.10 *$P(n;2,s) = P(n;3,s)$, for all $s = 4,\ldots,n+1$.*

Proof: The proof is by induction. The statement is true for $n \leq 8$ by inspection. Assume the truth of the theorem for n. Let $s \geq 4$. The coefficient $P(n+1;2,s)$ is equal to

$$
\begin{aligned}
&\sum_{i=1}^{s-1} P(n;i,s-1) + \sum_{j=s}^{n+1} P(n;2,j) \\
&= 0 + \sum_{i=2}^{s-1} P(n;i,s-1) + \sum_{j=s}^{n+1} P(n;2,j) \qquad (14)
\end{aligned}
$$

while $P(n+1;3,s)$ is

$$\sum_{i=2}^{s-1} P(n;i,s-1) + \sum_{j=s}^{n+1} P(n;3,j).$$

In these two formulæ, the first sums can be seen to be identical, while the second sums are termwise equivalent by the induction hypothesis. □

Proposition 3.5.11 *$P(n;r,r+1) = P(n;r,r+2)$, for all $r = 2,\ldots,n-1$.*

Proof: The proof is by induction. The statement is true for $n \leq 8$ by inspection. Assume the truth of the theorem for n. The coefficient $P(n+1; r, r+1)$ is equal to

$$\sum_{i=r-1}^{r} P(n; i, r) + \sum_{j=r+1}^{n+1} P(n; r, j)$$

$$= P(n; r-1, r) + 0 + P(n; r, r+1) + \sum_{j=r+2}^{n+1} P(n; r, j)$$

while $P(n+1; r, r+2)$ is

$$\sum_{i=r-1}^{r+1} P(n; i, r+1) + \sum_{j=r+2}^{n+1} P(n; r, j)$$

$$= (P(n; r-1, r+1) + P(n; r, r+1) + 0) + \sum_{j=r+2}^{n+1} P(n; r, j)$$

In these two formulæ, the sums can be seen to be identical, while the terms $P(n; r-1, r+1)$ and $P(n; r-1, r+1)$ are equal by the induction hypothesis, since if $2 \leq r \leq n$ then $1 \leq r-1 \leq n-1$. We apply the induction hypothesis unless $r - 1 = 1$, in which case both terms are 0. $\square$

We thus establish that the first two terms to the right of the diagonal are equal in every row except the nth. For row n, we have determined the values of each of the two terms explicitly in 3.5.5 and 3.5.7.

The two nonzero terms of row n also break another pattern which can be observed in the matrices $P(n)$ for $n \geq 5$. Every row except the nth is weakly decreasing from left to right. Likewise, every column is weakly decreasing. Actually, the rows and columns are strictly decreasing, except for the cases of equality treated in the two above propositions. We can establish by induction that once this pattern is established, it persists.

Theorem 3.5.12 *For $n \geq 5$, the above-diagonal elements of $P(n; r, s)$ are weakly decreasing along row r for $2 \leq r \leq n-1$. Except for the first row of zeros, the elements of $P(n; r, s)$ are weakly decreasing down each column.*

Proof: We give a proof by induction. We have already observed the truth of the theorem for $n = 5$. Next, assume the theorem holds for n. We check first that the columns of $P(n+1)$ are decreasing, by showing that $P(n+1; r, s) \leq P(n+1; r-1, s)$ if $3 \leq r < s \leq n+1$.

Appealing to the recursion formula of equation (10), we have

$$\begin{aligned}
P(n+1;r,s) &= \sum_{i=r-1}^{s-2} P(n;i,s-1) + \sum_{j=s}^{n+1} P(n;r,j) \\
P(n+1;r-1,s) &= \sum_{i=r-2}^{s-2} P(n;i,s-1) + \sum_{j=s}^{n+1} P(n;r-1,j) \\
&= P(n;r-2,s-1) + \sum_{i=r-1}^{s-2} P(n;i,s-1) + \sum_{j=s}^{n+1} P(n;r-1,j)
\end{aligned}$$

Comparing the two righthand sides, we find that the second begins with an extra nonnegative term, that the first sums are identical, and that each term of the second sum is greater than or equal to the corresponding term of the first sum, by the inductive hypothesis. This completes the first induction, in which we made use only of the decreasing columns of $P(n)$.

It remains to show that the rows of $P(n+1)$ decrease. In this induction, we will make use of both the decreasing rows and the decreasing columns of $P(n)$. We will show that $P(n+1;r,s) \leq P(n+1;r,s+1)$ if $2 \leq r < s \leq n+1$. We again appeal to equation (10).

$$\begin{aligned}
P(n+1;r,s) &= \sum_{i=r-1}^{s-2} P(n;i,s-1) + \sum_{j=s}^{n+1} P(n;r,j) \\
&= \sum_{i=r-1}^{s-2} P(n;i,s-1) + P(n;r,s) + \sum_{j=s+1}^{n+1} P(n;r,j) \qquad (15) \\
P(n+1;r,s+1) &= \sum_{i=r-1}^{s-1} P(n;i,s) + \sum_{j=s+1}^{n+1} P(n;r,j) \\
&= \sum_{i=r-1}^{s-2} P(n;i,s) + P(n;s-1,s) + \sum_{j=s+1}^{n+1} P(n;r,j) \qquad (16)
\end{aligned}$$

Again comparing the righthand sides (15) and (16), we see that the final sums are identical. Also, that each term of the first sum in (15) is at least as great as the corresponding term of the first sum in (16), as the rows of $P(n)$ are known to decrease by induction. And, $P(n;r,s) > P(n;s-1,s)$ by the induction hypothesis for the columns of $P(n)$. □

The statement about the rows in theorem 3.5.12 enables us to give a lower bound on the row sums $R(n;r)$, since the smallest entry in row r will be the one on the right, which we have determined in theorem 3.5.5. We multiply the least entry by the number of nonzero entries in row r.

Corollary 3.5.13 *The number of* 1234*-avoiding permutations of* $1, 2, \ldots, n$ *with initial decreasing subsequence of length* $r - 1$*, namely* $R(n; r)$*, is at least*

$$(n - (r - 1)) \cdot \left[\binom{2n - r}{n - r + 1} - \binom{2n - r}{n - r} \right]$$

The determination of the entries on the first nonzero diagonal would enable us to give a corresponding upper bound, as well as a lower bound for the column sums $C(n; s)$.

4 Introduction to stack-sortable permutations

4.1 The sorting algorithm

In the remainder of this thesis we shall be concerned with the operation of sorting permutations by passing them through a *first in, last out stack*. The numbers enter the stack in the order in which they occur in the input permutation, $\pi = (p_1, p_2, \ldots, p_n)$. We would like the output permutation to be the identity, $\iota = (1, 2, 3, \ldots, n)$.

The nature of the stack imposes certain restrictions. Namely, if $i < j$ then p_i is added to the stack before p_j; furthermore, p_i is removed from the stack either before p_j is added or after p_j is removed, but not between these two events.

If, then, we think of writing an open parenthesis every time an element is added to the stack, and a closed parenthesis every time one is removed, it is evident that the resulting sequence of parentheses is well-formed. For consider the pair of brackets belonging to p_i and the pair belonging to p_j; either the two pairs do not overlap, or one encloses the other.

Conversely, a well-formed sequence can be characterized by the property that closed parentheses never outnumber open parentheses counting from left to right. This property implies that the sequence of sorting operations indicated by a well-formed sequence can always be applied to a permutation, for we will never attempt to remove from the stack more elements than have yet been placed on it.

In section 2.1, we called a well-formed sequence of parentheses a *bracketing sequence*, and denoted the set of all bracketing sequences with n open and n closed parentheses by B_n.

To each element $b \in B_n$, we can associate a unique element of S_n which is put in order by the action of the bracketing sequence b. To find it, we label the closed parentheses in order $1, 2, \ldots, n$. These correspond to removing the elements $1, 2, \ldots, n$ in order from the stack. But if an element k is removed from the stack when a certain) is processed, then it must have been placed on the stack when the matching (was encountered. So we can also label the open parentheses; reading these off in order from left to right tells us the order in which the elements were placed on the stack.

For reference, we repeat the example from the end of section 2.1.

Example 4.1.1 *The bracketing sequence ()(()(())) sorts the permutation (1,5,2,4,3):*

$$\begin{array}{cccccccccc}
 & 1 & & & 2 & & & 3 & 4 & 5 \\
(&) & (& (&) & (& (&) &) &) \\
1 & & 5 & 2 & & 4 & 3 & & &
\end{array}$$

In section 2.1 we remarked that a permutation is stack-sortable if and only if it belongs to $S(231)$, i.e. if it avoids the pattern 231. From here on, it will be more convenient to refer to the set $S_n(231)$ as $1SS_n$, the set of *one-stack sortable permutations of length n.*

Definition 4.1.2 *If a permutation $\pi \in 1SS_n$, we say the element of B_n to which it is uniquely associated is its* sorting sequence, $b(\pi)$.

We can determine the sorting sequence associated to $\pi \in 1SS_n$ by inverting the process of example 4.1.1 above. Inspection of that example suggests the procedure. First, write an open parenthesis for each element of π in order from left to right. Second, process each from left to right. When an open parenthesis is processed, close all lower-numbered parentheses which are presently open, begininning with the leftmost. (In the example, when 4 is processed, 2 is closed, but 5 is not.) Third, when the final open parenthesis has been processed, close all remaining open parentheses, again beginning with the leftmost.

This interpretation characterizes stack-sorting in terms of bracketing sequences. In terms of a first-in, last-out stack, this process converts to the following. At each stage, compare the element presently on top of the stack to the next element to be added. If the element to be added is larger than the element on top of the stack, remove the top element; if it is smaller, add it to the stack. If the stack is empty, add to it; if the input is empty, clear the stack. The final condition of empty input corresponds to the third step in the preceding paragraph.

If an element k is removed from the stack as a consequence of a comparison with a larger element h being processed as input, we say that k is *removed by h*.

The sorting algorithm we have described in the two previous paragraphs is in essence a greedy algorithm. When the algorithm is presented in this way, it can readily be applied to arbitrary permutations in S_n, not merely to those in $1SS_n = S_n(231)$. Since we apply our greedy sorting algorithm to a permutation in an online fashion, we will not know whether the permutation is indeed one-stack sortable until we are finished, or forced to remove two elements out of order. So it makes sense to think of applying the algorithm to an arbitrary permutation and then examining the output. In fact, the input could be an arbitrary *permutation sequence*, i.e. an ordered list of n distinct integers, not necessarily the integers $1, 2, 3, \ldots, n$.

We can consider the operation of sorting, then, to be a function mapping an input permutation to an output permutation, $\Pi : S_n \to S_n$. In this interpretation, the set $1SS_n$ is just the preimage of the identity, ι_n, in S_n. If σ is an arbitrary permutation sequence, then $\Pi(\sigma)$ is well defined, and is a permutation sequence comprised of the same elements as σ.

The definition of sorting sequence, definition 4.1.2, also carries over to arbitrary permutation sequences.

Example 4.1.3 *If* $\pi = (1,5,2,4,3)$, *then* $\Pi(\pi) = (1,2,3,4,5)$.
Therefore, $\pi \in 1SS_5$. *The sorting sequence* $b(\pi) = ()(()(()))$.

If $\rho = (3,5,2,4,1)$, *then* $\Pi(\rho) = (3,2,1,4,5)$.
Therefore, $\rho \notin 1SS_5$. *The sorting sequence* $b(\rho) = ()(()(()))$.

If $\sigma = (4,2,6,7,1)$, *then* $\Pi(\sigma) = (2,4,6,1,7)$.
The sorting sequence $b(\sigma) = (())()(())$.

The idea of a sorting function, Π, is a useful one, but should be considered with a grain of salt. There is no easier way to determine the action of Π on an arbitrary permutation in S_n than to execute the steps of the sorting algorithm.

One reason for considering the function Π, is the facility it gives us for making the following definition. We say that a permutation π is *two-stack sortable* if and only if $\Pi^2(\pi) = \Pi(\Pi(\pi)) = \iota$. That is, we pass the permutation through the stack, executing the greedy sorting algorithm. Then we take the output and pass *it* through the stack, again executing the same algorithm. Since $\Pi(\iota) = \iota$, every one-stack sortable permutation is also two-stack sortable.

We can extend the definition to k passes through a stack.

Definition 4.1.4 *The set of k-stack sortable permutations is defined as follows.*

$$kSS_n = \{\pi \in S_n : \Pi^k(\pi) = \iota\}$$

Every permutation which is $(k-1)$-stack sortable is k-stack sortable; that is, $(k-1)SS_n \subseteq kSS$

Fundamental problems in this area are to characterize the members of kSS_n and to enumerate the k-stack sortable permutations. We might also ask which and how many permutations are k-stack sortable but not $(k-1)$-stack sortable; to this end we make the following definition.

Definition 4.1.5 *The set of* exactly *k-stack sortable permutations is defined as follows.*

$$\Delta kSS_n = kSS_n \setminus (k-1)SS_n = \{\pi \in S_n : \Pi^{k-1} \in 1SS_n \setminus \iota\}$$

If $\pi \in S_n$ is given by $\pi = (a_1, a_2, \ldots, a_{k-1}, n, b_1, b_2, \ldots, b_{n-k})$, and $\alpha = (a_1, a_2, \ldots, a_{k-1})$, $\beta = (b_1, b_2, \ldots, b_{n-k})$, then we write $\pi = \alpha n \beta$. In abbreviating a permutations in this fashion, we use greek letters for permutations and permutation sequences, and reserve roman letters and arabic numerals for individual elements of a permutation.

Lemma 4.1.6 *If for $\pi \in S_n$, $\pi = \alpha n \beta$, then $\Pi(\pi) = \Pi(\alpha)\Pi(\beta)n$.*

Proof: Consider the application of the sorting algorithm to π. When the element n is reached, all the elements of α and none of β have been processed. Some may remain on the stack. The element n is larger than every element on the stack, and so the stack is cleared. Thus the elements of α are output as $\Pi(\alpha)$, exactly as though an end-of-input had been reached. Next the element n is entered onto the stack. As it is larger than every element of β, n remains on the stack until the end-of-input is reached. So n does not interfere with the processing of β, which is output as $\Pi(\beta)$. Finally, an end-of-input is reached, and n is removed from the stack. □

Porism 4.1.7 *If $\rho = \Pi(\pi)$ for any $\pi \in S_n$, then $\rho(n) = n$.*

Proof: Since n appears at some position in π, π can always be written in the form $\pi = \alpha n \beta$ for some α, β. □

Since after one pass, the largest element has been shifted to the end, two passes will shift the largest two elements to the end, and so on. We thus give a more general version of the porism.

Porism 4.1.8 *If $\rho = \Pi^k(\pi)$ for any $\pi \in S_n$, then $\rho(j) = j$, for $n - k + 1 \leq j \leq n$.*

Proof: The proof is by induction. The statement is vacuously true for $k = 0$, and true for $k = 1$ by porism 4.1.7.

If $\rho = \Pi^{k+1}(\pi)$, then $\rho = \Pi(\Pi^k(\pi))$. By the induction hypothesis, $\Pi^k(\pi)$ has its k largest elements in order in the final k positions. When the first of these is encountered, it will clear the stack, being larger than any previous input. The rest of the elements are encountered in increasing order, and so are simply passed through.

So if $\Pi^k(\pi) = (a_1, a_2, \ldots, a_{n-k}, n-k+1, \ldots, n)$, we can take $\alpha = (a_1, a_2, \ldots, a_{n-k}) \in S_{n-k}$. By the remarks of the previous paragraph, $\Pi^{k+1}(\pi) = \Pi(\Pi^k(\pi)) = \Pi(\alpha(n-k+1)\ldots(n-1)(n)) = \Pi(\alpha)(n-k+1)\ldots(n-1)(n)$. By porism 4.1.7, $\Pi(\alpha)$ has $n-k$ for its final element. Hence $\Pi^{k+1}(\pi)$ has $(n-k, n-k+1, \ldots, n-1, n)$ for its final elements. □

These observations lead to one important result, namely that the sorting process always terminates.

Theorem 4.1.9 $\Pi^n(\pi) = \iota_n$ *for all* $\pi \in S_n$.

Proof: This is simply porism 4.1.8 with $k = n$. □

Actually, $n-1$ iterations will do, for if the last $n-1$ positions of a permutation of length n are occupied by $2, 3, \ldots, n$, then clearly the first element is 1.

Since the sorting algorithm requires $2n$ steps to process a permutation of length n, namely one step to put each element on the stack, and one step to take each off, with a comparison performed at each step, the algorithm requires $O(n)$ steps. In theorem 4.1.9, we have just seen that $O(n)$ repeated applications of the algorithm suffice to sort any sequence.

Corollary 4.1.10 *The stack-sorting algorithm is an* $O(n^2)$ *sorting algorithm.*

We will see in the following section that $\Omega(n^2)$ steps are actually required in the worst case.

We make one further remark about the function Π, which amounts to an alternate proof that the sorting procedure terminates.

Definition 4.1.11 *Let* $<_L$ *denote the lexicographic order on* S_n. *That is,* $\sigma <_L \tau$ *if and only if there exists a* k *for which* $\sigma(j) = \tau(j)$ *for* $j = 1, 2, \ldots, k-1$ *and* $\sigma(k) < \tau(k)$.

Proposition 4.1.12 *If* $\pi \in S_n$, $\pi \neq \iota_n$, *then* $\Pi(\pi) <_L \pi$.

Proof: First suppose that $\Pi(\pi) = \pi$. Then, since $\Pi(\pi)(n) = n$, $\pi(n) = n$. By the same token, $\pi(n-1) = n-1$, etc. So it is easy to see that $\Pi(\pi) = \pi$ if and only if $\pi = \iota$. That is, the identity is the only permutation fixed by the function Π.

Now take $\pi \neq \iota$, and let $\rho = \Pi(\pi)$. Then $\rho \neq \pi$. Let k be the first index for which $\rho(k) \neq \pi(k)$. Denote $\rho(k)$ by r and $\pi(k)$ by p. Evidently r has left the stack before p, since the element p has not yet appeared in ρ. But p was placed on the stack first, as the element r has not yet appeared in π. So r was placed on the stack on top of p. This would not have happened unless $r < p$.

Hence $\rho <_L \pi$. □

As the elements of S_n are totally ordered by $<_L$, it follows that the sequence π, $\Pi(\pi)$, $\Pi^2(\pi), \ldots$ is strictly decreasing until $\Pi^k(\pi) = \iota$ for some k.

This approach does not, however, give us the stronger result of theorem 4.1.9, namely that n passes will suffice.

Example 4.1.13 *If* $\rho = (3, 5, 2, 4, 1)$, *then* $\Pi(\rho) = (3, 2, 1, 4, 5)$. ρ *and* $\Pi(\rho)$ *differ in their second position, and* $2 < 3$, *so* $\Pi(\rho) <_L \rho$.

If $\sigma = (3, 2, 1, 4, 5)$, *then* $\Pi(\sigma) = (1, 2, 3, 4, 5)$. $\Pi(\sigma)$ *is the lexicographically least permutation, the identity. Therefore* $\Pi(\sigma) <_L \sigma$.

4.2 The sorting tree

In this section, we introduce a rooted tree whose covering relations are defined by the sorting function Π. This model facilitates the formulation of questions about the sorting function.

For every n, we define a directed graph with $n!$ nodes, labelled by the elements of S_n. For each $\pi \in S_n$, we place an edge from π to $\Pi(\pi)$. Since the function Π is well defined, every node has outdegree 1. If we delete the edge which runs from ι_n to $\iota_n = \Pi(\iota_n)$, consideration of either 4.1.9 or 4.1.12 makes it clear that there are no cycles in the remaining graph.

It follows that the graph is actually a tree, with all edges directed towards a root, labelled by the identity permutation, ι_n. We denote this tree by $T(n)$. We consider the root to be on level 0, and a node whose unique path to the root has length k to be on level k.

The one-stack sortable permutations are just those which label nodes connected directly to the root, namely the nodes on level one. Similarly, π is *exactly k-stack sortable*, $\pi \in \Delta kSS_n$, precisely when π appears on level k. The *k-stack sortable* permutations are those that appear on or above level k.

Thus our fundamental questions about the stack-sorting operation transfer naturally to questions about the sorting tree $T(n)$. When does a permutation appear on or above the k-th level? What are the level sums of the tree? What is the depth of the tree? &c.

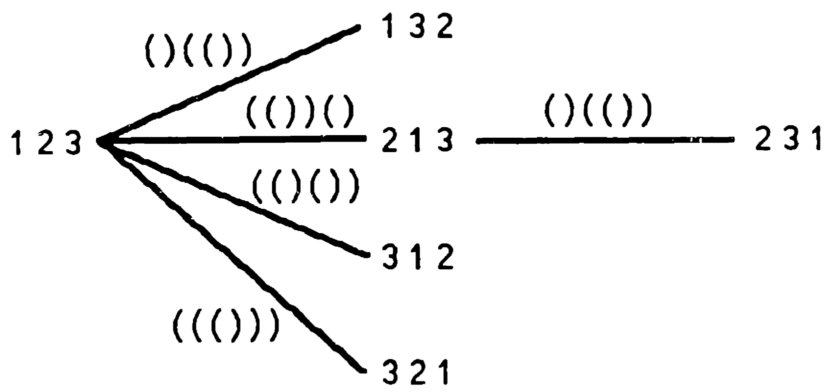
It is also convenient to think of the *edges* of $T(n)$ as being labelled with bracketing sequences. Label the edge connecting π to its parent $\Pi(\pi)$ with the sorting sequence $b(\pi)$.

The trees $T(3)$ and $T(4)$ are depicted in the accompanying figure. It might be remarked that $T(n)$ has depth $n-1$ at least for $n = 3, 4$. Theorem 4.1.9 guarantees that the depth is at most $n-1$ for all n; in these cases the bound is obtained.

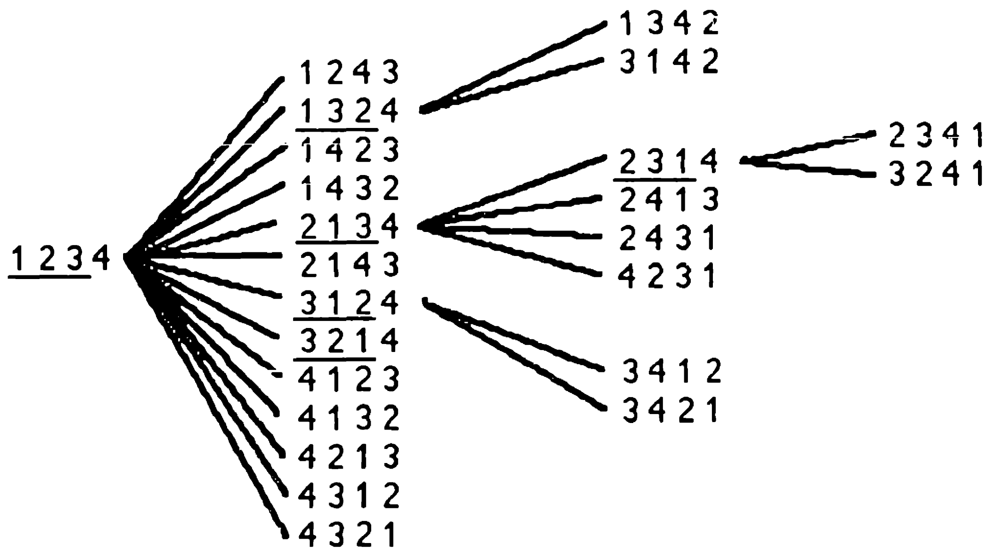
Notice that if $\pi \in S_n$ has the form $\pi = \pi^* n$ for some $\pi^* \in S_{n-1}$, then $\Pi(\pi) = \Pi(\pi^*)n$ and $b(\pi) = b(\pi^*)()$.

Therefore, a structure isomorphic to $T(n-1)$ can be found embedded in $T(n)$, as the elements with $\pi(n) = n$. The nodes in question in $T(4)$ have been underlined in the figure, and can be seen to form a copy of $T(3)$.

Indeed, we can regard every finite permutation belonging to $\cup_{n=1}^{\infty} S_n$ as an infinite permutation in an obvious way. If $\pi \in S_n$, let $\pi(k) = k$ for $k > n$. In this sense, every tree $T(n)$ is a restriction of an ideal infinite tree $T(\infty)$ whose nodes are labelled by the infinite permutations. But since our questions involve describing the structure of the finite trees $T(n)$, this does not seem to be a particularly helpful point of view.



The sorting tree T(3)



The sorting tree T(4), with elements of T(3) underlined

4.2.1 Some basic lemmata

In this section we settle for certain cases one of the fundamental questions raised in section 4.1 and reformulated in the preceding section. When does a permutation $\pi \in S_n$ appear on the k-th level of the tree $T(n)$?

We give a characterization reminiscent of the forbidden subsequences discussed in the first half of this thesis.

Recall from chapter 1 that a *wedge* in a permutation $\pi \in S_n$ is a subsequence of length three indexed by $j_2 < j_3 < j_1$ with $\pi(j_1) < \pi(j_2) < \pi(j_3)$. For instance, in the permutation $\pi = (3,5,2,4,1)$, the elements 3, 4 and 1 form a wedge. In this case, $j_2 = 1$, $j_3 = 4$, and $j_1 = 5$. We check that $\pi(5) < \pi(1) < \pi(4)$.

As remarked in section 2.1, the one stack sortable permutations are precisely those which contain no wedges. Hence, a permutation appears *below* level one in $T(n)$ if and only if it contains a wedge.

The results of this section will generalize this characterization. Before proceding, we prove two basic lemmata.

We will say that an element p precedes an element q in a permutation ρ if $\rho^{-1}(p) < \rho^{-1}(q)$. For instance, in $\rho = (3,5,2,4,1)$, the element 5 precedes 4 because $\rho^{-1}(5) = 2$ and $\rho^{-1}(4) = 4$.

Lemma 4.2.1 *If $\pi \in S_n$, and $1 \leq a < b \leq n$, and if a precedes b in π then a precedes b in $\Pi(\pi)$.*

Proof: Since a precedes b in π, a enters the stack before b. When b is processed, either a has already been removed from the stack, in which case a will precede b in $\Pi(\pi)$, or a must be removed from the stack to accomodate the larger element b. □

Lemma 4.2.2 *If $\pi \in S_n$, and $1 \leq a < b \leq n$, and if b precedes a in π, then b precedes a in $\Pi(\pi)$ if there exists $c > b$ such that b precedes c and c precedes a in π. If there is no such c, a precedes b in $\Pi(\pi)$.*

Proof: If there is a c satisfying the given conditions, then b, c and a form a wedge. In this case, b must be removed from the stack before c is placed on. Since c is placed on the stack before a, this will cause b to precede a in $\Pi(\pi)$.

On the other hand, if there is *no* c satisfying the given conditions, then b remains on the stack until a is processed. Since $a < b$, a will be placed on the stack above b, and so a precedes b in $\Pi(\pi)$. □

Porism 4.2.3 *If b and a form an inversion in $\Pi(\pi)$, that is if b precedes a in $\Pi(\pi)$ but $b > a$ then there is a wedge b, c, a in π for some $c > b$.*

Proof: An easy consequence of the two lemmata. If $b > a$, either b precedes a in π or *vice versa*. Only in the case that b precedes a and a larger element c is interposed between the two might b precede a in $\Pi(\pi)$. $\square$

4.2.2 Tree depth and the bottom levelsum

In theorem 4.1.9 we observed that every permutation in S_n was fully sorted by $n-1$ applications of the sorting algorithm. In this section, we prove that for every n there are some permutations which actually require $n-1$ passes through the stack. At the same time, we find the number of these permutations.

Recall the notation of section 4.1 in which we write permutations as concatenated strings, using greek letters for permutation sequences and roman letters and arabic numerals for individual elements. For instance, we observed in a porism that for all $\pi \in S_n$ it is the case that $\Pi(\pi) = \alpha n$ for some $\alpha \in S_{n-1}$. Since then $\Pi^k(\pi) = \Pi^{k-1}(\Pi(\pi)) = \Pi^{k-1}(\alpha n) = \Pi^{k-1}(\alpha)n$, it is true that π is exactly k-stack sortable if and only if α is exactly $(k-1)$-stack sortable.

We use this observation in the following inductive proof.

Theorem 4.2.4 *$\pi \in \Delta(n-1)SS_n$ if and only if $\pi = \rho n 1$ for some $\rho \in S_{n-2}$.*

Proof: The statement is true for $n = 3$ as $\Delta 2 SS_3 = \{231\}$. (It is also true for $n = 2$.)

Now assume the truth of the given statement for $n-1$. A permutation $\pi \in S_n$ is exactly $(n-1)$-stack sortable if and only iff $\Pi(\pi) = \alpha n$ where $\alpha \in S_{n-1}$ is exactly $(n-2)$-stack sortable. We check that likewise π has the form $\rho n 1$ if and only if α has the form $\beta, (n-1), 1$.

The proof will then follow by induction. The two classes, of permutations having the given form, and of permutations requiring the maximum number of passes to sort, are equivalent for $n-1$ by the induction hypothesis. The arguments of the previous paragraph will show them also to be equivalent for n.

First let $\pi = \rho n 1$. Then $\Pi(\pi) = \Pi(\rho)1n$. Since $\Pi(\rho)$ will have the form $\sigma, n-1$, we can write $\Pi(\pi) = \alpha n$, where $\alpha = \sigma, n-1, 1$. This is the desired form.

Conversely, suppose

$$\Pi(\pi) = \sigma, n-1, 1, n$$

Write π in the form $\pi_L n \pi_R$, so that

$$\Pi(\pi) = \Pi(\pi_L)\Pi(\pi_R)n$$

Since both $\Pi(\pi_L)$ and $\Pi(\pi_R)$ must end with an ascent if they have length greater than 1, a comparison of the forms 4.2.2 and 4.2.2 reveals that $\Pi(\pi_R) = 1$. Then $\pi = \pi_L n 1$, the desired form. □

Corollary 4.2.5 $|\Delta(n-1)SS_n| = (n-2)!$

Proof: This is an immediate consequence of the above characterization of the elements of $\Delta(n-1)SS_n$. □

Corollary 4.2.6 *For $n \geq 1$, the tree $T(n)$ has depth $n-1$.*

Proof: We know from theorem 4.1.9 that the depth is at most $n-1$. Since $\Delta(n-1)SS_n$ is never empty for $n \geq 1$, there are in fact elements on level $n-1$. □

Corollary 4.2.7 *Sorting according to the greedy stacksorting algorithm requires $\Omega(n^2)$ time in the worst case.*

Proof: In theorem 4.1.9 we observed that $n-1$ passes sufficed in all cases. Now, conversely, we have established that $n-1$ passes are sometimes necessary. □

Having succeeded in characterizing the permutations on the bottom level of the sorting tree $T(n)$, we next seek to generalize this to other levels. We will give a necessary condition for a permutation to be below the kth level, and another condition which is sufficient. These will lead to a precise characterization of the elements on the next-to-bottom level, and to a determination of the levelsum for the penultimate level.

4.2.3 A necessary, and a sufficient, condition

Definition 4.2.8 *A forbidden pattern of order k in a permutation π is a pair of elements $j_2 < j_1$, with $\pi(j_1) < \pi(j_2)$, together with a set A of k elements, with $\pi^{-1}(a) < j_2$ and $\pi(j_1) < a < \pi(j_2)$ for each $a \in A$.*

Example 4.2.9 *A forbidden pattern of order* 1 *is simply a wedge.*

The pair of elements $(4,1)$, *together with the set* $\{2,3\}$, *forms a forbidden pattern of order* 2 *in each of the permutations* $(2,3,4,1)$ *and* $(3,2,4,1)$.

We first prove that if a permutation contains no forbidden pattern of order k, then it appears on or above level k in the sorting tree. Therefore, every permutation appearing below the kth level contains a forbidden pattern of order k.

Theorem 4.2.10 *If* $\pi \in S_n$ *contains no forbidden pattern of order* k, *then* $\Pi^k(\pi) = \iota_n$.

Proof: The proof is by induction on k. As a forbidden pattern of order 1 is simply a wedge, and since the wedge-free patterns are exactly those appearing on levels 0 and 1 of $T(n)$, the theorem holds for $k = 1$.

It will suffice to show that if $\pi \in S_n$ contains no forbidden pattern of order k, then $\Pi(\pi)$ contains no forbidden pattern of order $k-1$. The proof will then follow by induction. We prove the contrapositive statement, that if $\Pi(\pi)$ *does* contain a forbidden pattern of order $k-1$, then π contains a forbidden pattern of order k.

Let a forbidden pattern of order $k-1$ in $\Pi(\pi)$ be comprised of the pair $c > a$ (where c precedes a) with the set $A = \{b_1, \ldots, b_{k-1}\}$. Since c is greater than and precedes a in $\Pi(\pi)$, there must be some wedge (c, x, a) in π, where $x > c$. Now, if any b_i fell to the right of x in π, then (c, x, b_i) would also be a wedge in π, so that c would precede b_i in $\Pi(\pi)$. As this is not the case, each member of A, as well as c, must precede x in π. Therefore the pair $x > a$, together with the set $A \cup c$, forms the required forbidden pattern of order k in π. □

Example 4.2.11 *The permutation* $\pi = (2,5,3,4,1)$ *contains no forbidden pattern of order* 3. *It appears on level* 3 *of* $T(5)$, *as* $\Pi(\pi) = (2,3,1,4,5)$; $\Pi^2(\pi) = (2,1,3,4,5)$; *and* $\Pi^3(\pi) = (1,2,3,4,5)$.

The permutation $\rho = (3,5,2,4,1)$ *contains no forbidden pattern of order* 3. *It appears on level* 2 *of* $T(5)$, *as* $\Pi(\pi) = (3,2,1,4,5)$; *and* $\Pi^2(\pi) = (1,2,3,4,5)$.

The first permutation in this example appears below level 2, thus the contrapositive statement to the theorem says that it must contain a forbidden pattern of order 2, as it does: the pair $(4,1)$ and the set $\{2,3\}$ form such a pattern. This pair and this set also form a pattern of order 2 in the permutation of the second example, which serves to show that having a forbidden pattern of order k is not sufficient to consign a permutation to a position below the kth level.

Having thus given a condition which is necessary but not sufficient to locate a permutation below level k, we next give a sufficient condition.

Definition 4.2.12 *A forbidden pattern of order k, comprised of a pair $c > a$ and a set of elements B preceding c will be called* uninterrupted *if there is no subsequence (b_1, x, b_2) in π, where $x > c$, for any $b_1, b_2 \in B$.*

Definition 4.2.13 *A forbidden pattern of order k, comprised of a pair $c > a$ and a set of elements B preceding c will be called* pure *if there is no subsequence (b_1, x, b_2) in π, where $x > max_{b \in B} b$, for any $b_1, b_2 \in B$.*

If a permutation contains an uninterrupted forbidden pattern of order k which is not pure, then there exists a subsequence of the form (b_1, x, b_2) in π, where $max_{b \in B} b < x < c$. Then x can be appended to the set B to form an uninterrupted forbidden pattern of one higher order. The uninterrupted forbidden pattern of maximal order in π is therefore always pure. We shall be concerned precisely with this highest-order pattern.

Theorem 4.2.14 *If $\pi \in S_n$ contains an uninterrupted forbidden pattern of order k, then π appears below level k in the sorting tree $T(n)$. That is, $\Pi^k(\pi) \neq \iota_n$.*

Proof: As in the proof of the previous theorem, we fix n and offer an induction on k. The theorem is true for $k = 1$, as any permutation containing a pure uninterrupted forbidden pattern of order 1, in other words a wedge, is not one-stack sortable and so appears below level 1.

Now suppose that $\pi \in S_n$ contains an uninterrupted forbidden pattern of order k, and therefore a pure uninterrupted forbidden pattern of order $h \geq k$. We claim that $\Pi(\pi)$ contains an uninterrupted forbidden pattern of order $h - 1$; therefore, by the induction hypothesis, $\Pi(\pi)$ requires at least h passes to sort, so π requires at least $h + 1$.

Let the pure forbidden pattern of order k in π be composed of the pair $c > a$ and the set B. Consider the application of the sorting procedure Π to π. By the purity of the forbidden pattern, no member of the set B will be removed until all members of B have been placed on the stack. Then, when an element larger than $b_{\max} = \max_{b \in B} b$ is encountered, all members of B remaining on the stack are removed before the larger element is placed on. Only this larger element can remove $b_{\max}$, so that $b_{\max}$ will be the rightmost member of B in the sorted permutation $\Pi(\pi)$. Also, since $(b_{\max}, c, a)$ is a wedge in π, $b_{\max}$ precedes a in $\Pi(\pi)$.

Therefore the pair $b_{\max} > a$ and the set $B \setminus \{b_{\max}\}$ form an uninterrupted forbidden pattern in $\Pi(\pi)$. □

Example 4.2.15 *Each of the $(n-2)!$ permutations on level $n-1$ of the tree $T(n)$ contains a forbidden pattern of order $n-2$. In this extreme case, each element in the permutation contributes to the forbidden pattern. There are no other permutations containing such a forbidden pattern; accordingly, all other permutations appear above level $n-1$.*

Example 4.2.16 *Each of the forbidden patterns of order $n-2$ seen in example 4.2.15 to appear in the permutations on the bottom level of $T(n)$, level $n-1$, is pure. Accordingly, these permutations appear below level $n-2$.*

We can also use these two theorems to characterize, and so enumerate, the permutations of the bottom *two* levels; that is, those permutations $\pi \in \Delta(n-2)SS(n) \cup \Delta(n-1)SS(n)$. By theorem 4.2.10, if a permutation has no forbidden subpattern of order $n-3$, then it appears on or above level $n-3$. Hence the permutations we seek all have forbidden subpatterns of order $n-3$. Such a pattern, comprised of a pair $c > a$ and a set B, is pure *unless* $c = n-1$, $a = 1$, $B = \{2, \ldots, n-2\}$. Therefore, every permutation having a forbidden pattern of order $n-3$ actually appears on level $n-2$ or below, with the possible exception of permutations having the form $\pi = (\ldots, n-1, 1)$. We examine these permutations more closely.

If $\pi = (\ldots, n-1, 1)$ and π contains a wedge of the form $(n-2, n, x)$, then the element $(n-2)$ will precede x in $\Pi(\pi)$. Since $(x, n-1, 1)$ is also a wedge in π, it is also the case that the element x will precede the element 1 in $\Pi(\pi)$. Also, the rightmost two elements of $\Pi(\pi)$ will be $(n-1, n)$ as neither n nor $n-1$ is removed from the stack until the end of the permutation is reached. As at least the four elements $x, 1, n-1, n$ all succeed $n-2$ in $\Pi(\pi)$, no more than $n-5$ elements might precede $n-2$. Therefore $n-2$ cannot be the element c in a forbidden pattern of order $n-4$ in the permutation $\Pi(\pi)$. Also, as no smaller element succeeds $n-1$ or n, neither can participate in a forbidden pattern of order $n-4$. Therefore, $\Pi(\pi)$ is free of patterns of order $n-4$, and so is $(n-4)$-stack sortable. Since then π is $(n-3)$-stack sortable, π does not lie on the bottom two levels of $T(n)$.

On the other hand, suppose π contains no wedge of the form $(n-2, n, x)$. Then $n-2$ is removed from the stack by $n-1$. The sorted permutation $\Pi(\pi)$ therefore has the form $(\ldots, n-2, 1, n-1, n)$. By inspection, this contains a pure forbidden pattern of order $n-3$, and so $\Pi(\pi)$ appears on or below level $n-3$. But, also by inspection, $\Pi(\pi)$ contains no forbidden pattern of order $n-2$, and so appears above level $n-2$. Clearly, $\Pi(\pi) \in \Delta(n-3)SS_n$, and so $\pi \in \Delta(n-2)SS_n$.

We can thus characterize the permutations of the bottom two levels as being all those containing a forbidden pattern of order $n-3$, with the exception of permutations of the form

$(\ldots, n-2, \ldots, n, \ldots, x, \ldots, n-1, 1)$. (We have already determined those on the bottom level to be just the permutations of the form $(\ldots, n, 1)$, in the previous subsection.) We next use this characterization to enumerate the permutations of level $n-2$.

Theorem 4.2.17 *The number of permutations in $\Delta(n-2)SS(n)$ is $\frac{7}{2}(n-2)! + (n-3)!$*

Proof: A permutation containing a forbidden pattern of order $n-3$ has one of the following forms, where the x's represent arbitrary elements:

$$
\begin{array}{llll}
(p, \ldots & \ldots & p, n, 1) & \\
(p, \ldots & \ldots & p, n-1, 1) & (17) \\
(p, \ldots & p, & n, 1, p) & \\
(p, \ldots & p, & n, p, 1) & (18) \\
(p, \ldots & \ldots & p, n, 2) & \\
(p, \ldots & p, & n-1, 1, n) &
\end{array}
$$

Permutations of the first form also contain a forbidden pattern of order $n-2$, and are the $(n-2)!$ permutations of $\Delta(n-1)SS(n)$. We are interested in permutations of the other five forms. The last of these has $n-3$ undetermined elements p and so corresponds to the $(n-3)!$ permutations. The other four have $n-2$ undetermined elements, and so there are $(n-2)!$ permutations of each of these forms. Only a permutation of the form (17) might be of the exceptional form In exactly half of the permutations of form (17), the element n will precede the element $n-2$. These do not have the exceptional form and so belong to $\Delta(n-2)SS(n)$. In the other half, $n-2$ precedes n. These permutations have the exceptional form, except for permutations of the form $(\ldots, n-2, \ldots, n, n-1, 1)$. These have already been counted among the permutations of form 18. So exactly half of the permutations of form (17) are to be counted. We obtain a total of $\frac{1}{2}(n-2)! + (n-2)! + (n-2)! + (n-2)! + (n-3)$ permutations. □

This formula, and the formula for the bottom levelsum, can be compared to the data in the following table, which give $\Delta kSS(n)$ for all $n \leq 11$:

$k\backslash n$	1	2	3	4	5	6	7	8	9	10	11
0	1	1	1	1	1	1	1	1	1	1	1
1		1	4	13	41	131	428	1429	4860	16 794	58 784
2			1	8	49	276	1509	8184	44 473	243 334	1 343 654
3				2	23	198	1556	11 812	88 566	662 732	4 975 378
4					6	90	982	9678	91 959	863 296	8 093 662
5						24	444	5856	68 820	775 134	8 618 740
6							120	2640	40 800	555 828	7 201 188
7								720	18 360	325 200	5 033 952
8									5040	1461 60	2 918 160
9										40 320	1 310 400
10											362 880

4.2.4 A conjecture about level 2

We give a characterization, similar to those of the previous subsection, for the two-stack sortable permutations.

Theorem 4.2.18 *A permutation $\pi \in S_n$ fails to be two-stack sortable if it contains a subsequence of type* 2341, *or a subsequence of type* 3241 *which is not part of a subsequence of type* 35241. *If it contains no such subsequence, π is two-stack sortable.*

Proof: The proof is an exercise in the application of the basic lemmata 4.2.1 and 4.2.2.

First suppose π has a subsequence of either of the given forms, and consider $\Pi(\pi)$. First consider a subsequence of type 2341, consisting of the elements b, c, d, a where $a < b < c < d$. Since b precedes c in π and $b < c$, it follows that b will precede c in $\Pi(\pi)$, regardless of the other elements of π. Also, because c, d, a form a wedge in π, c will precede a in $\Pi(\pi)$. Therefore, the elements b, c, a appear in that order in $\Pi(\pi)$, where they form a wedge. Since therefore $\Pi(\pi)$ is not one-stack sortable, it follows that π is not two-stack sortable.

Second, consider a subsequence of type 3241, say consisting of the elements c, b, d, a, where there is no element larger than d which follows c but precedes b. There are two cases: either there is an element $x > c$ which follows c but precedes b, or there is no such element. If there is such an x, by assumption $c < x < d$, and consequently c, x, d, a is of type 2341 and we are back in the case of the preceding paragraph. Otherwise, if there is no such element $x > c$, then b precedes c in $\Pi(\pi)$. And since c, d, a is a wedge in π, c precedes a in $\Pi(\pi)$. Once again, b, c, a form a wedge in $\Pi(\pi)$.

It follows that if π has one of the forbidden subsequences, then π fails to be two-stack sortable.

Conversely, we can show that if $\Pi(\pi)$ fails to be one-stack sortable; that is, if it contains a wedge, then π must contain one of the two forbidden subsequences.

Suppose that b, c, a form a wedge in $\Pi(\pi)$. We look at two cases; either b precedes c in π or *vice versa.*

If b precedes c in π, then porism 4.2.3 guarantees a wedge c, x, a in π. But then b, c, x, a is a subsequence of type 2341.

If c precedes b in π, then there can be no $x > c$ such that c precedes x and x precedes b in π. But since c precedes a in $\Pi(\pi)$, there is some wedge c, y, a in π. Since $y > c$, y cannot precede b in π, by the remark in the first sentence of this paragraph. Therefore b precedes y, and c, b, y, a is a sequence of type 3241. Again by the remark in the first sentence, this subsequence is not part of a subsequence of type 35241.

Thus if b, c, a is a wedge in $\Pi(\pi)$, we see that π has a subsequence of one of the two forms given in the statement of the theorem. So if π is not two-stack sortable, it has one of the forbidden subsequences. □

The above theorem does not, strictly speaking, give a characterization in terms of forbidden subsequences. It does not permit us to write the class $2SS_n$ as an intersection of sets of the form $S_n(\tau)$, because of the unfamiliar restriction that "forbidden" subsequences of type 3241 are permitted if each is mitigated by being part of a 35241. Nevertheless, it has much the same flavour of our characterization of $1SS_n$ as $S_n(231)$.

We should like to exploit this knowledge about $2SS_n$ to enumerate this class of permutations. An equivalent problem would be to count $\Delta 2SS_n = 2SS_n \setminus 1SS_n$, as we know that $|1SS_n| = |S_n(()|231) = c_n$, the Catalan number. Unfortunately, we do not know of a method which would permit this calculation. Several possibilities are explored in the sequel. But the data in the table given in the previous subsection suggest a simple closed form.

Conjecture 4.2.19 $|2SS_n| = \frac{2}{(n+1)(2n+1)}\binom{3n}{n}$.

Comparing the form of this conjecture with the expression for the Catalan numbers, $\frac{1}{n+1}\binom{2n}{n}$ would suggest that an expression could be found for $3SS_n$ of the form $\frac{1}{P(n)}\binom{4n}{n}$, where $P(n)$ is a low degree polynomial in n. But the data in the table above do not support such a conjecture for any $P(n)$ of degree less than 7.

4.3 Unasked questions

The sorting function Π and the trees $T(n)$ derived from it arise from an attempt to model the problem of sorting by passage through a series of stacks. The algorithm which generates the sorting function Π has considerable advantages. It is very simple to implement, and is independent of the length of the input, so that it is not even necessary to know how many, or which, elements are in the input. It might, however, be fairly objected that the function Π is a very naïve sorting algorithm, and some less rigid model of sorting on a stack might be considered. In this thesis, we have not considered allowing any more powerful operation than the naïve, greedy algorithm.

But in the interest of equal time, we present a series of alternative formulations of the stack-sorting question. None of these questions has, to the best of our knowledge, received serious attention.

Question 4.3.1 *Can a better algorithm be designed for stacks which are not the last in a series?*

If we have only one stack, then the best sorting algorithm, in the sense of the one which achieves success for the greatest number of inputs, is the greedy algorithm. But if the output from a stack is to be sorted again, a better algorithm can be devised. For instance, suppose we have two stacks and we know that the input is of length 4. If both stack operate according to the greedy algorithm, the two inputs $(3,2,4,1)$ and $(2,3,4,1)$ will not be sorted after passing through the second stack. But suppose the first stack implements a different algorithm, so that if the first two elements to be input are 3 and 2, the *larger* is output first (and otherwise operates as the greedy algorithm). Then the stacks will sort as follows:

input	output from first stack	output from second stack
2, 3, 1, 4	3, 1, 2, 4	1, 2, 3, 4
2, 3, 4, 1	3, 2, 1, 4	1, 2, 3, 4
3, 2, 1, 4	3, 1, 2, 4	1, 2, 3, 4
3, 2, 4, 1	3, 2, 1, 4	1, 2, 3, 4

So this implementation of a two-stack system is more successful than using two copies of the greedy algorithm. But we have made use of a foreknowledge of a length of the input, as well as which elements are in the input permutation. If the first stack sees that the first two inputs are 2 and 3 (in some order), then it also knows that the inputs to follow are 4 and 1.

If a stack knows where it is located in a series of stacks, it can exploit this knowledge, without any further communication with the other stacks, as we have just seen. As well as using the information that the output will be sent on to more stacks, a stack could use the fact that its input has already been passed through one or more stacks, as the output from a stack has a special form, being already sorted. For instance, if a stack is the rth in a series, it might be able to assume that it is operating on an input of length $n - r$, if the stacks which precede it are each using an algorithm that sends one more large element to the end of the permutation.

Perhaps even more information than this can be used. For instance, if the first stack in a series runs the greedy algorithm on an input of length 6, there are only 68 possible outputs, down from $6! = 720$ inputs. This is even considerably smaller than the upper bound of $5! = 120$ obtained from the fact that the largest element has been sent to the end.The second stack thus has relatively few possible inputs with which to contend. If a stack knew that it was following a stack operating the greedy algorithm, it would be forewarned to expect one of these 68 permutations as input.

Question 4.3.2 *Is there an efficient algorithm which makes use of the all the data which has already been passed?*

The variant of the greedy algorithm which was described above suggested deviating from the greedy algorithm if two specific elements, 2 and 3, were being compared as the top element on the stack and the next element of the input. However, this deviation was only planned if this was the first comparison being made. We could imagine an algorithm which made use, not only of how much input has already been seen, but of what that input was (and, since the algorithm is deterministic, it would also implicitly know what output has resulted). Again, the situation breaks into two cases, depending on whether or not the stack has been forewarned about which elements are in the input. If so, it would also have knowledge about what elements remained, although not about the order in which they were to appear.

Although this algorithm is considerably more sophisticated than the naïve algorithm, it still operates in an online fashion. We could abandon this requirement, as in the following question.

Question 4.3.3 *How well can we do if the entire input is studied before sorting begins?*

If the algorithm controlling a stack were allowed to scan the entire input permutation before making *any* decisions, it would become very powerful indeed. In effect, this amounts to the non-deterministic version of the naïve algorithm. Rather than try to select the most efficient way to process an input, we just try all possible ways of processing the input at each stack, and declare the operation to be successful if *any one* of the possibe outputs from the final stack is fully sorted. (The final stack in a series can continue to run the greedy algorithm, as we know that this is the most efficient in this special case.)

Rather than one option at each stack, as in the naïve, deterministic case, we have c_n possible ways to process an input of length n. Therefore, the simple upper bound on the number of permutations of length n sortable by a series of k nondeterministic stacks is $(c_n)^k$, which is asymptotically equal to $4^{n \cdot k} = (4^k)^n$. Although this is eventually outstripped by $n!$ for any k, it happens quite slowly. For instance, for 2 nondeterministic stacks, the simplistic upper bound only guarantees that some permutation of length $n = 41$ is not sortable. (As $16^{40} = 1.5 \times 10^{48}$ and $40! = 8.2 \times 10^{47}$).

However, running a nondeterministic algorithm on an input of this length clearly requires absurd amounts of time and storage. Even the idea of scanning a permutation of length 40 and making a decision about how to sort it requires a significant amount of memory compared to the very simple, memoryless action of the naïve sorting algorithm. Although we have not been concerned about efficient computation in this thesis, it might still be of interest to ask how well we can do with a prescribed finite amount of memory attached to each stack. For instance, we might be able to remember, and make decisions based on, all of the elements presently in the stack, instead of just the topmost.

Another arrangement halfway between working online and nondeterminism would be to make the first stack operated online, but allow the remainder full knowledge of their input, the permutation having already been scanned once.

Question 4.3.4 *Is there any advantage to occasional random deviation from the greedy algorithm?*

If we think of all the permutations of length n as being arrayed in the sorting tree $T(n)$, then the sorting algorithm Π steps resolutely up the tree from child to parent. It seems possible that by occasionally making random transitions we might have more likelihood of making a sudden advance up the tree than of suffering a sudden setback. This would amount to shaking out a knot in a permutation all at once, instead of letting a small element progress forwards through a permutation one step at a time.

We suggest two ways in which we could make random transitions. First, we could decide with some small probability to apply a random bracketing sequence to an input permutation, rather than the specific one prescribed by the sorting algorithm. This random bracketing sequence could be chosen either with uniform probability, or according to the some function of how commonly it appears in the sorting tree. (The question of how many sequences require a given bracketing sequence in the operation of the greedy algorithm is taken up in section 5.4.) Secondly, we could operate the naïve algorithm, but each time a comparison is made between a stack element and an input, to obey the algorithm with probability $(1 - p)$ and perform the reverse action with probability p. In each of these two cases, it seems possible that there is some nonzero probability which would confer the greatest advantage.

5 More about the sorting tree

Fleurs sorties des parenthèses d'un pas
– Louise de Vilmorin,
Fiançailles pour rire

5.1 The average depth

We would like to determine the average depth of the sorting tree $T(n)$. That is, given a uniformly selected random permutation of S_n, how many times does the sorting algorithm have to be applied before the permutation is completely sorted? We make use of the characterizations of subsection 4.2.3, and ask what is the highest order forbidden pattern which can be expected in a permutation of length n? We will prove the following, not very ambitious, result.

Theorem 5.1.1 *For large n, the sorting tree $T(n+1)$ has average depth of at least $.23n - .62\sqrt{n}$.*

Proof: Recall from subsection 4.2.3 that a permutation falls below level k if it contains an uninterrupted forbidden pattern of order k.

We consider all permutations in S_{n+1} and the depth in $T(n+1)$ of each one. First, we throw out any contribution from half the permutations, and retain that half which in which the largest element, $n+1$, falls between positions $n/2 - \sqrt{n}$ and $n - \sqrt{n}$. If the largest element of a permutation is serving as the element c in a forbidden pattern of order k, that pattern will always be uninterrupted. We therefore look for a high order forbidden pattern using $n+1$ as the element c.

If the element $n+1$ falls in the given range, then there are at least $\sqrt{n}$ positions to its right. The chances are high that one of the elements $1, 2, \ldots, \sqrt{n}$ appears in one of these positions. Consider filling a permutation of length n in such a way that none of $1, 2, \ldots, \sqrt{n}$ appears in one of the rightmost $\sqrt{n}$ positions. Place the small elements first, in $(n - \sqrt{n})(n - \sqrt{n} - 1) \cdots (n - 2\sqrt{n} + 1)$ ways. Then place the remaining elements in the $(n - \sqrt{n})$ open positions. This gives in total

$$(n - \sqrt{n})(n - \sqrt{n} - 1) \cdots (n - 2\sqrt{n} + 1) \cdot (n - \sqrt{n})!$$

permutations. How does this number compare to the total number $n!$ of permutations of length n? The ratio is

$$\left(\frac{n-\sqrt{n}}{n}\right)\left(\frac{n-\sqrt{n}-1}{n-1}\right)\cdots\left(\frac{n-2\sqrt{n}+1}{n-\sqrt{n}+1}\right)\left(\frac{(n-\sqrt{n})!}{(n-\sqrt{n})!}\right)$$

$$\begin{aligned}
&\leq \left(\frac{n-2\sqrt{n}}{n-\sqrt{n}}\right)^{\sqrt{n}} \\
&= \left(\frac{n-\sqrt{n}-\sqrt{n}}{n-\sqrt{n}}\right)^{\sqrt{n}} \\
&= \left(1-\frac{\sqrt{n}}{n-\sqrt{n}}\right)^{\sqrt{n}} \\
&= \left(1-\frac{1}{\sqrt{n}-1}\right)^{\sqrt{n}} \\
&\leq \frac{1}{e}
\end{aligned}$$

Thus, the majority of these permutations, at least $(1-\frac{1}{e}) > .63$ of them, have an element of size at most $\sqrt{n}$ which can act as the element a in a forbidden pattern. We consider now only these permutations, which make up $\frac{1}{2}(1-\frac{1}{e}) > .31$ of the total. Since the element $n+1$, acting as c, is preceded by, on average, $(\frac{3n}{4}-\sqrt{n})$ elements, and since all but at most $\sqrt{n}$ of these elements are larger than the element acting as a, there is a forbidden pattern of average order $(\frac{3n}{4}-2\sqrt{n})$ for these permutations.

We use only that each of the other .69 of the permutations has depth greater than or equal to zero. This gives an average depth for the tree of at least

$$(.31)(\frac{3n}{4}-2\sqrt{n}) = (.23n) - (.62\sqrt{n})$$

□

Clearly, we have cut corners in order to simplify the proof. In particular, by taking the last $c \cdot \sqrt{n}$ for some $c > 1$, we can increase the coëfficient on the dominant linear term at the expense of the squareroot term. This should bring the leading term arbitrarily close to $\frac{3}{8}n$. Also, when $n+1$ is near the middle of the permutation, we have not made use of the fact that it is followed by considerably more than $\sqrt{n}$ positions. And we have drawn no advantage at all from fully half the permutations. For those permutations where $n+1$

is very close to the front of the permutation, we might choose to let n take its place as the element c in the forbidden pattern.

However, the theorem we have just proved does give a lower bound with a linear term, and so is sufficient for the purpose of the following corollary.

Corollary 5.1.2 *Sorting according to the greedy stacksorting algorithm requires $\Omega(n^2)$ time in the average case.*

Proof: In corollary 4.2.7 we concluded a worst case time of $\Omega(n^2)$ from the fact that the total depth of the tree was linear. We have now concluded that a linear number of passes is required on average to sort a permutation. □

This establishes conclusively that the naïve sorting algorithm is not computationally efficient. There are many familiar algorithms which sort in $O(n \log n)$ time.

We speculate that the bound of the above theorem can be improved to $\frac{n}{2} - c \cdot \sqrt{n}$, by using the result on uninterrupted forbidden patterns; to do any better, we would have to use a more precise characterization of the permutations on the kth level. We would like to complement this result with an upper bound on the average depth, derived from the complementary result of theorem 4.2.10 on (ordinary) forbidden patterns. Unfortunately, this will not get us near $\frac{n}{2}$, as forbidden patterns are far more common if we drop the requirement that they be uninterrupted.

We have seen that a larger proportion, $(1 - \frac{1}{e})$, of all permutations of length n are such that one of the $\sqrt{n}$ smallest elements falls somewhere in the rightmost $\sqrt{n}$ places. More generally, if $c > 1$ is a parameter, then $(1 - \frac{1}{e^c})$ of all permutations have one of the $c \cdot \sqrt{n}$ smallest elements in one of the rightmost $\sqrt{n}$ places. Furthermore, of these the same proportion $(1 - \frac{1}{e^c})$ will have one of the largest $c \cdot \sqrt{n}$ elements in one of the next $\sqrt{n}$ places from the right. This small element and this large element form the elements a and c in a forbidden pattern of order at least $n - (2c+2)\sqrt{n}$. This is so as there are $n - 2\sqrt{n}$ positions to the left of our large element $c > n - c\sqrt{n}$, and at most $2c\sqrt{n}$ of these positions are occupied by an element either larger than c or smaller than $a \leq c\sqrt{n}$.

It follows that the average depth of the tree cannot be shown by avoidance of forbidden patterns of order k to be any less than

$$(1 - \frac{1}{e^c}) \cdot (1 - \frac{1}{e^c}) \cdot (n - (2c + 2)\sqrt{n})$$

Here by taking a large value of the parameter c we can make the coëfficient of n arbitrarily close to 1 at the cost of making the coëfficient of the squareroot term prohibitively large. The actual average depth lies somewhere between the two bounds of $.23n - .62\sqrt{n}$ and $n - c_1 \cdot \sqrt{n}$. We speculate that it is nearer to the latter than the former, and that in fact the correct average depth is probably of the form $(1-\epsilon)n - c_1 \cdot \sqrt{n}$, for some constant c_1. To determine this, it will be necessary to use a more precise argument than the characterizations in terms of forbidden patterns and of uninterrupted forbidden patterns, which are far apart. In addition to showing the average depth to be of the form $c_2 \cdot n - c_1 \cdot \sqrt{n}$ and explicitly determining c_2, it would be good to determine what fraction of the nodes in the tree lies below level $c_2 \cdot n - c_1 \cdot \sqrt{n}$, where the parameter c_1 is allowed to vary.

5.2 Results on fertility

In this section, we concern ourselves further with the structure of the sorting trees $T(n)$ introduced in chapter 4. In particular, we are concerned with the number of children of a given node.

Definition 5.2.1 *Let $Ch(\pi)$ denote the set of children of a the node labelled π in $T(n)$. The fertility of the node π, $f(\pi)$, is the number of children of the node. That is,*

$$\begin{aligned} Ch(\pi) &= \{\rho \in S_n : \Pi(\rho) = \pi\} \\ f(\pi) &= |Ch(\pi)| \end{aligned}$$

We will also be interested in certain subtrees of $T(n)$. Let $T(\rho)$ denote the subtree of $T(n)$ consisting of the node labelled ρ together with all its descendents. Thus $T(\iota_n) = T(n)$. Also, if $\rho \in S_n$ and $\rho(n) \neq n$, then, in view of porism 4.1.7, $T(\rho)$ consists of the single node labelled by ρ.

Recall that the Catalan numbers are specified by $c_0 = 1$ and the recurrence 1:

$$c_n = \sum_{j=1}^{n} c_{j-1} c_{n-j}, \tag{19}$$

which has the solution given in equation 2:

$$c_n = \binom{2n}{n} - \binom{2n}{n-1} = \frac{1}{n+1}\binom{2n}{n}. \tag{20}$$

We use the Catalan numbers to define the following two-parameter family:

Definition 5.2.2 *For $n \geq 3$ and $2 \leq k \leq n-1$, define*

$$d_{n,k} = \sum_{j=2}^{k} c_{j-1} c_{n-j} \tag{21}$$

Letting $k = 2$ in definition 5.2.2, we obtain the identity

$$d_{n,2} = c_{n-2}.$$

The following basic relation also follows directly from the definition.

$$d_{n,k} = d_{n,k-1} + c_{k-1} \cdot c_{n-k} \qquad (3 \leq k \leq n-1)$$

And by comparison of equations (19) and (21), observe that

$$d_{n,n-1} = c_n - 2 \cdot c_{n-1}.$$

The values of $d_{n,k}$ for $n \leq 7$ are given in the following table.

n	$d_{n,2}$	$d_{n,3}$	$d_{n,4}$	$d_{n,5}$	$d_{n,6}$
3	1				
4	2	4			
5	5	9	14		
6	14	14	34	48	
7	42	70	95	123	165

Definitions 5.2.3 *For $n \geq 3$ and $2 \leq k \leq n-1$, we define three permutations, $\mu_{n,k}, \nu_{n,k}, \xi_{n,k} \in S_n$ as follows.*

$$\begin{aligned}
\mu_{n,k} &= (2, 3, \ldots, k, 1, k+1, \ldots, n) \\
\nu_{n,k} &= (1, 2, \ldots, k-2, k, k-1, k+1, \ldots, n) \\
\xi_{n,k} &= (k, 1, 2, \ldots, k-1, k+1, \ldots, n)
\end{aligned}$$

We remark that $\mu_{n,2} = \nu_{n,2} = \xi_{n,2} = (2,1,3,4,\ldots,n)$. We also remark, although we will not make use of, the facts that ${\mu_{n,k}}^{-1} = \xi_{n,k}$ and ${\nu_{n,k}}^{-1} = \nu_{n,k}$.

Notice the following relations which serve to locate each μ, ν and ξ in the sorting tree $T(n)$.

$$\begin{aligned}
\Pi(\mu_{n,k}) &= \mu_{n,k-1} \\
\Pi(\mu_{n,2}) &= \iota_n \\
\Pi(\nu_{n,k}) &= \iota_n \\
\Pi(\xi_{n,k}) &= \iota_n
\end{aligned}$$

The first result we prove is the following:

Theorem 5.2.4 *$f(\xi_{n,n+1-k}) = d_{n,k}$, for all $n \geq 3$, $2 \leq k \leq n-1$.*

Proof: The proof is by induction on k. For the base case, we check that $f(\xi_{n,n-1}) = d_{n,2} = c_{n-2}$. Suppose that $\Pi(\rho) = \xi_{n,n-1}$. Write ρ in the form $\rho = \alpha n \beta$, and note that $\Pi(\alpha)\Pi(\beta) = (n-1, 1, 2, \ldots, n-2)$. Since the largest element in this permutation sequence is $(n-1)$, and since both $\Pi(\alpha)$ and $\Pi(\beta)$ must end with their largest elements, it follows that $\Pi(\alpha) = (n-1)$ and $\Pi(\beta) = (1, 2, \ldots, n-2)$. There is thus only one possibility for α, and as the requirement for β is precisely that $\beta \in 1SS_{n-2}$, there c_n possibilities for β. The permutation $\xi_{n,n-1}$ thus has c_n children in $T(n)$.

To complete the induction, we wish to show that for all $3 \leq k \leq n-1$, if $f(\xi_{n,n+1-k-1}) = d_{n,k-1}$, then

$$f(\xi_{n,n+1-k}) = d_{n,k} = d_{n,k-1} + c_{k-1} \cdot c_{n-k}.$$

We will partition $Ch(\xi_{n,n+1-k})$ into two sets, one counted by $d_{n,k-1}$ and the other by $c_{k-1} \cdot c_{n-k}$. The first set will be the set of all $\rho \in Ch(\xi_{n,n+1-k})$ such that the element $n-k+1$ is removed by $n-k+2$ when ρ is sorted. For permutations in the second set, $n-k+1$ will be removed by some larger element.

Note that $\xi_{n,n+1-k}(1) = \xi_{n,n+2-k}(n+2-k) = n+1-k$, that $\xi_{n,n+1-k}(n+2-k) = \xi_{n,n+2-k}(1) = n+2-k$, and that $\xi_{n,n+1-k}(j) = \xi_{n,n+2-k}(j)$ for all other indices j. In other words, $\xi_{n,n+1-k}$ can be obtained from $\xi_{n,n+2-k}$ by interchanging the positions of the elements $(n+1-k)$ and $(n+2-k)$.

We now claim that if $\rho \in Ch(\xi_{n,n+2-k})$, and if σ is obtained from ρ by interchanging the positions of the two elements $(n+1-k)$ and $(n+2-k)$, then $\sigma \in Ch(\xi_{n,n+1-k})$. Moreover, $n+1-k$ is not removed by $n+2-k$ when σ is sorted. The key observation is that since

$n+1-k$ and $n+2-k$ differ by one, they interact identically with all other elements, both the larger and the smaller.

In $\Pi(\rho) = \xi_{n,n+2-k}$, the element $(n+2-k)$ precedes the element $(n+1-k)$. Therefore, $(n+2-k)$ must have been removed from the stack before $(n+1-k)$ was encountered. Hence the two elements do not interact at all. But they interact identically with all other elements. Therefore, if their positions are interchanged to form the permutation σ, then σ and ρ are associated to the same bracketing sequence by the application of the sorting algorithm.

Thus $\Pi(\sigma)$ is obtained from σ by precisely the same permutation by which $\Pi(\rho)$ is obtained from ρ. It follows that $\Pi(\sigma)$ differs from $\Pi(\rho)$ only in that the elements $(n+2-k)$ and $(n+1-k)$ are interchanged. That is, $\sigma \in Ch(\xi_{n,n+1-k})$.

On the other hand, if σ is a permutation such that $\Pi(\sigma) = \xi_{n,n+1-k}$ and if $n+1-k$ is not removed by $n+2-k$, then we can interchange these two elements to obtain a permutation ρ such that $\Pi(\rho) = \xi_{n,n+2-k}$.

The remaining elements of of $Ch(\xi_{n,n+1-k})$ will be those for which $n+1-k$ *is* removed by the element $n+2-k$. An inspection of $\xi_{n,n+1-k} = (n+1-k, 1, 2, \ldots, n-k, n+2-k, \ldots, n)$ will reveal certain restrictions on a permutation π having this property. First, obviously $n+1-k$ must precede $n+2-k$ in π. Second, if $j > n+2-k$ then j cannot fall between $n+1-k$ and $n+2-k$ in π, for then $n+1-k$ would not be removed by $n+2-k$.

Now consider an element $h < n+1-k$, h cannot precede $n+1-k$ in π, for then h would have to precede $n+1-k$ in $\Pi(\pi)$, which is not the case. In fact, h could not fall after $n+1-k$ and precede $n+2-k$, for then there would be no element larger than $n+1-k$ between $n+1-k$ and h, so h would again precede $n+1-k$ in $\Pi(\pi)$. It follows that the elements $n+1-k$ and $n+2-k$ are in fact *adjacent* in π. Moreover, they are followed immediately by all of the $n-k$ elements smaller than $n+1-k$. For each of these elements is to the right of $n+2-k$ in π. If any element x larger than $n+2-k$ preceded any element h smaller than $n+2-k$, then the three elements $(n+2-k, x, h)$ would form a wedge in π, and so $n+2-k$ would precede h in $\Pi(\pi)$, which it does not.

These $n-k$ smallest elements must be arranged in the order of a stack-sortable permutation, as they appear in exact ascending order in $\xi_{n,n+1-k}$. They can be so arranged in c_{n-k} ways. So arranged, the elements $1, 2, \ldots, n+1-k, n+2-k$ form a block in π. That block will be processed as a unit and lead to the output substring $(n+1-k, 1, 2, \ldots, n-k, n+2-k)$. This must be the *first* substring of the output $\xi_{n,n+1-k}$. If we replace this block of the $n+2-k$ smallest elements by a *single* element 0, we obtain a permutation sequence comprised of the $k-1$ elements $0, n+3-k, \ldots, n$.

It must be the case that when this permutation sequence is sorted the elements are output in ascending order: the block 0 first, followed by the rest in ascending order, just as they appear in $\xi_{n,n+1-k}$. There are c_{k-1} ways to make this arrangement.

Since for each of the c_{n-k} arrangements of the smallest elements there are c_{k-1} arrangements of the rest of the elements, there are in total $c_{n-k} \cdot c_{k-1}$ permutations in $Ch(\xi_{n,n+1-k})$ such that $n+1-k$ is removed by $n+2-k$. This completes the induction step. □

Example 5.2.5 *In $T(5)$ the children of the node $\xi_{5,4} = (4,1,2,3,5)$ are labelled by the $d_{n,2} = 5$ permutations:*

$$\begin{matrix} 4 & 5 & 1 & 2 & 3 \\ 4 & 5 & 1 & 3 & 2 \\ 4 & 5 & 2 & 1 & 3 \\ 4 & 5 & 3 & 1 & 2 \\ 4 & 5 & 3 & 2 & 1 \end{matrix}$$

The node $\xi_{5,3} = (3,1,2,4,5)$ has $d_{n,3} = d_{n,2} + c_2 \cdot c_2 = 5 + 2 \cdot 2 = 9$ children. Five of these are obtained from the children of $(4,1,2,3,5)$ by interchanging the positions of elements 3 and 4:

$$\begin{matrix} 3 & 5 & 1 & 2 & 4 \\ 3 & 5 & 1 & 4 & 2 \\ 3 & 5 & 2 & 1 & 4 \\ 3 & 5 & 4 & 1 & 2 \\ 3 & 5 & 4 & 2 & 1 \end{matrix}$$

The rest are obtained by forming the $c_2 = 2$ blocks $(3,4,1,2)$ and $(3,4,2,1)$, and combining each with the remaining element 5 in c_2 ways:

$$\begin{matrix} (3412)5 \\ (3421)5 \\ 5(3412) \\ 5(3421) \end{matrix}$$

We next prove a parallel result for $\mu_{n,k}$.

Theorem 5.2.6 *$f(\mu_{n,k}) = d_{n,n-k+1}$ for all $n \geq 3$ and $2 \leq k \leq n-1$.*

Proof:

Recall the form of $\mu_{n,k}$:

$$\mu_{n,k} = (2,3,\ldots,k,1,k+1,\ldots,n).$$

Let $\pi \in Ch(\mu_{n,k})$ so that $\Pi(\pi) = \mu_{n,k}$. It is evident by the observations of section 4.2.1 that if s is one of the elements $2, 3, \ldots, k$, then s precedes the element 1 in π, and furthermore there is a wedge of the form $(s, x, 1)$ in π.

Now, define $t = \pi(j)$ to be the largest (equivalently, leftmost) element so that $t = \pi(j) > \pi(j+1) > \cdots > \pi(h) = 1$.

Clearly, there is such a t, or 1 would be the first element of $\Pi(\pi)$. Furthermore, $t \geq k+1$, as if t were one of the elements $2, 3, \ldots, k$ there would be no wedge $(t, x, 1)$ in π.

We form the partition

$$Ch(\mu_{n,k}) = \mathcal{P}_{k+1} \cup \mathcal{P}_{k+2} \cup \cdots \cup \mathcal{P}_n$$

where $\pi \in \mathcal{P}_t$ if t is the largest element so that $t = \pi(j) > \pi(j+1) > \cdots > \pi(h) = 1$.

We proceeded to characterize, and so enumerate, the sets $\mathcal{P}_t$. In what follows, in addition to n and k we can regard t as being fixed.

We now consider sorting a permutation $\pi \in \mathcal{P}_t$. Because of the way in which we have selected t, all of the elements between t and 1 follow the element t onto the stack before any element is removed. The element 1 is of course removed from the stack immediately. Since 1 follows each of the elements $2, 3, \ldots, k$ in $\Pi(\pi) = \mu n, k$, this means that each of these elements precedes t in π.

Now let s be any of the elements $k+1, k+2, \ldots, t-1$. The element s must not precede t in π, otherwise $(s, t, 1)$ would form a wedge, and s would precede 1 in $\Pi(\pi)$, which is not the case. But there can be no element $x > t$ forming a wedge (t, x, s) in (π), for then s would succeed t in $\Pi(\pi)$, which is neither the case. And since the smaller elements $2, 3, \ldots, k$ all precede t, it must be the case that the $t-k$ elements 1 and $k+1, k+2, \ldots, t-1$ together form a block which immediately succeeds t in π. The next element that follows this block is larger than t and so will have the effect of clearing the stack at least down to the level of the element t. The elements of the block are to appear in ascending order in $\Pi(\pi) = \mu_{n,k}$, and consequently can appear in exactly c_{t-k} ways within the block.

The elements outside this block, excepting t, number $n - (t - k) - 1$. They are $2, 3, \ldots, k$ and $t+1, t+2, \ldots, n$. These elements are also to appear in ascending order in . Consequently they can be arranged into a list in any of exactly $c_{n-t+k-1}$ ways. It only remains to insert the element t and its trailing block into this list. But we know exactly where t must be located. The element immediately preceding t cannot be any of $t+1, t+2, \ldots, n$, because these elements are larger than t and would contradict the maximality of t. So it must be one of $2, 3, \ldots, k$. To be precise, it must be the rightmost of these, because we know that each of these elements precedes t in π.

So we are able to obtain the elements of $\mathcal{P}_t$ by following these steps: divide the elements into the sets $\{1, k+1, k+2, \ldots, t-1\}$ and $\{2, 3, \ldots, k, t+1, t+2, \ldots, n\}$. Form the first of these into a block in one of c_{t-k} ways, and the second into a list in one of $c_{n-t+k-1}$ ways. Insert each of the blocks, together with the element t, into each of the lists in a unique way. It follows that

$$|\mathcal{P}_t| = c_{t-k} \cdot c_{n-t+k-1}.$$

As $Ch(\mu_{n,k})$ is precisely the disjoint union $\bigcup_{t=k+1}^{n} \mathcal{P}_t$, we calculate

$$\begin{aligned} f(\mu_{n,k}) = |Ch(\mu_{n,k})| &= \sum_{t=k+1}^{n} c_{t-k} \cdot c_{n-t+k-1} \\ &= \sum_{t=2}^{n-k+1} c_{(t+k-1)-k} \cdot c_{n-(t+k-1)+k-1} \\ &= \sum_{t=2}^{n-k+1} c_{t-1} \cdot c_{n-t} \\ &= d_{n,n-k+1} \end{aligned}$$

□

Example 5.2.7 *We list the $d_{6,4} = 34$ children of $\mu_{6,3} = (2,3,1,4,5,6)$. The members of $\mathcal{P}_4$ are in the first column, those of $\mathcal{P}_5$ are in the second, and those of $\mathcal{P}_6$ are in the third. The block trailing the element t is bracketed in each case.*

23(41)56	23(514)6	
23(41)65	23(541)6	23(6145)
253(41)6		23(6154)
263(41)5	263(514)	23(6415)
2653(41)	263(541)	23(6514)
32(41)56		23(6541)
32(41)65	732(514)6	
523(41)6	32(541)6	
532(41)6		32(6145)
623(41)5	623(514)	32(6154)
6253(41)	623(541)	32(6415)
632(41)5		32(6514)
6523(41)	632(514)	32(6541)
6532(41)	632(541)	

A comparison of theorems 5.2.4 and 5.2.6 reveals the following.

Corollary 5.2.8 *$f(\mu_{n,k}) = f(\xi_{n,k})$ for all $n \geq 3$ and $2 \leq k \leq n-1$.*

Having succeeded in determining $f(\xi_{n,k})$ and $f(\mu_{n,k})$, and finding them to be equal for all appropriate choices for n and k, we turn naturally to the enumeration of $f(\nu_{n,k})$. The reader may not be surprised to find that $f(\nu_{n,k}) = d_{n,n+1-k}$ as well. In the following section we prove a stronger result, and obtain the desired enumeration as a corollary.

5.3 A subtree isomorphism

The goal of this section is to determine $f(\nu_{n,k})$, where $\nu_{n,k} = (1,2,\ldots,k-2,k,k-1,k+1,\ldots,n)$. Since we have already determined $f(\mu_{n,k})$, where $\mu_{n,k} = (2,3,\ldots,k,1,k+1,\ldots,n)$, it will suffice to prove the following theorem.

Theorem 5.3.1 *For all $n \geq 3$ and $2 \leq k \leq n-1$, we have the subtree isomorphism*

$$T(\mu_{n,2}) \cong T(\nu_{n,k})$$

Proof: We construct an explicit isomorphism.

Notice that $\mu_{n,k} = (2,3,\ldots,k,1,k+1,\ldots,n)$ can be obtained from $\nu_{n,k} = (1,2,\ldots,k-2,k,k-1,k+1,\ldots,n)$ by permuting the elements in the following way: add one to each element of value less than $k-1$, and replace $k-1$ by 1. We will prove that the same operation can be performed on each node of the subtree $T(\nu_{n,k})$ to obtain the nodes of the subtree $T(\mu_{n,k})$.

More formally, if $\pi \in S_n$, define the permutation $\rho = M_k^n(\pi)$ by $\rho(\pi^{-1}(j)) = \phi(j)$ where

$$\begin{aligned} \phi(j) &= j+1 && \text{for } 1 \leq j \leq k-2 \\ \phi(k-1) &= 1 && \\ \phi(j) &= j && \text{for } k \leq j \leq n \end{aligned} \tag{22}$$

As already remarked, $\mu_{n,k} = M_k^n(\nu_{n,k})$. We define the inverse mapping N_k^n similarly; if $\pi \in S_n$, define the permutation $\rho = N_k^n(\pi)$ by $\rho(\pi^{-1}(j)) = \psi(j)$ where

$$\begin{aligned} \psi(1) &= k-1 && \\ \psi(j) &= j-1 && \text{for } 2 \leq j \leq k-1 \\ \psi(j) &= j && \text{for } k \leq j \leq n \end{aligned} \tag{23}$$

Since the permutations ϕ and ψ, acting on the elements, are mutually inverse, the two mappings M_k^n and N_k^n are easily seen to be inverse for all choices of n and k it follows *a fortiori* that $\nu_{n,k} = N_k^n(\mu_{n,k})$.

Now fix n and k and let ν^* be any node in the subtree $T(\nu_{n,k})$. That is, there exists a unique positive integer r such that

$$\Pi^r(\nu^*) = \nu_{n,k}.$$

We claim that then

$$\Pi(M_k^n(\nu^*)) = M_k^n(\Pi(\nu^*)). \tag{24}$$

Conversely, if μ^* is a node of the subtree $T(\mu_{n,k})$, and if

$$\Pi^r(\mu^*) = \mu_{n,k}$$

for some $r \geq 1$, we claim that

$$\Pi(N_k^n(\mu^*)) = N_k^n(\Pi(\mu^*)). \tag{25}$$

From these two claims, it will follow directly that the mapping M_k^n induces an isomorphism from the subtree $T(\nu_{n,k})$ to the subtree $T(\mu_{n,k})$. This can be confirmed by straightforward induction on r.

Notice that if ν^* is a node of $T(\nu_{n,k})$ other than the root $\nu_{n,k}$, then $\nu' = \Pi(\nu^*)$ is also a node of $T(\nu_{n,k})$. Thus from the claim of equation 24 we derive

$$\begin{aligned}
\Pi(M_k^n(\nu')) &= M_k^n(\Pi(\nu')) \\
\Pi(M_k^n(\Pi(\nu^*))) &= M_k^n(\Pi(\Pi(\nu^*))) \\
\Pi(\Pi(M_k^n(\nu^*))) &= M_k^n(\Pi(\Pi(\nu^*))) \\
\Pi^2(M_k^n(\nu^*)) &= M_k^n(\Pi^2(\nu^*)).
\end{aligned}$$

Likewise, $\Pi^s(M_k^n(\nu^*)) = M_k^n(\Pi^s(\nu^*))$ for all s, in particular for $s = r$, so that $\Pi^r(M_k^n(\nu^*)) = M_k^n(\Pi^r(\nu^*)) = M_k^n(\nu_{n,k}) = \mu_{n,k}$. Therefore, the image under M_k^n of a node on the rth level of $T(\nu_{n,k})$ is a node on the rth level of $T(\mu_{n,k})$. It is no harder to see that the ancestry relations hold throughout, and the two trees are isomorphic.

It remains to verify the claims of (24) and (25). We first check (24).

For fixed n and k, let $\nu^* = (a_1, a_2, \ldots, a_n)$ be some node of $T(\nu_{n,k})$, and let ϕ be the permutation (depending on n and k) defined in (22). By definition, $M_k^n(\nu^*) = (\phi(a_1), \ldots, \phi(a_n))$. We claim that ν^* and $M_k^n(\nu^*)$ are associated to the same bracketing sequence by the action of the sorting algorithm. This can be seen to be identical to the condition that we seek to prove, namely that $M_k^n(\Pi(\nu^*)) = \Pi(M_k^n(\nu^*))$.

The permutation ϕ preserves all pairwise comparisons except that $\phi(1) > \phi(j)$ if $2 \leq j \leq k-1$. Therefore, when ν^* and $M_k^n(\nu^*)$ are sorted, exactly the same steps will be taken unless the element 1 is ever compared to any element j $(2 \leq j \leq k-1)$ in the sorting of ν^*. But this can never happen. If 1 were ever compared to such an element j, then 1 would precede j in $\Pi(\nu^*)$ and consequently in $\Pi^s(\nu^*)$ for all $s \geq 1$. In particular, 1 would precede j in $\Pi^r(\nu^*) = \nu_{n,k}$. Inspection of $\nu_{n,k}$ shows that this is not the case. Hence ν^* and $M_k^n(\nu^*)$ are associated with the same bracketing sequence.

We next check the assertion of (25).

For fixed n and k, let $\mu^* = (b_1, b_2, \ldots, b_n)$ be some node of $T(\mu_{n,k})$, and let ψ be the permutation (depending on n and k) defined in (23). By definition, $N_k^n(\mu^*) = (\psi(a_1), \ldots, \psi(a_n))$. As above, we claim that $\mu*$ and $N_k^n(\mu^*)$ are associated to the same bracketing sequence by the action of the sorting algorithm.

The permutation ψ preserves all pairwise comparisons except that $\psi(k-1) < \psi(j)$ if $1 \leq j \leq k-2$. Therefore, when μ^* and $N_k^n(\mu^*)$ are sorted, exactly the same steps will be taken unless the element $k-1$ is ever compared to any element j $(1 \leq j \leq k-2)$ in the sorting of μ^*. But this can never happen. If $k-1$ were ever compared to such an element j, then either j would be placed above $k-1$ on the stack, or j would be removed to output at the time $k-1$ was input. In this case, possibly some other elements, also smaller than $k-1$, are also removed at the same time. But in either case, not element larger than $k-1$ will be removed after j but before $k-1$. That is, all elements lying after j but before $k-1$ in $\Pi(\mu^*)$ are smaller than $k-1$. This pattern must then persist in $\Pi^s(\mu^*)$ for all $s \geq 1$, and, in particular, in $\Pi^r(\mu^*) = \mu_{n,k}$. Inspection of $\mu_{n,k}$ shows that this is not the case: in $\mu_{n,k}$, k, which is larger than $k-1$, precedes $k-1$ but follows every j for $1 \leq j \leq k-2$.

Hence μ^* and $M_k^n(\mu^*)$ are associated with the same bracketing sequence. □

Since if two nodes have isomorphic subtrees they certainly have the same number of children, an immediate consequence of the preceding theorem is the following.

Corollary 5.3.2 $f(\mu_{n,k}) = f(\nu_{n,k})$ *for all* $n \geq 3$, $2 \leq k \leq n-1$.

Combining corollary 5.3.2 with theorem 5.2.6 provides the desired result:

Corollary 5.3.3 $f(\mu_{n,n+1-k}) = f(\nu_{n,n+1-k}) = d_{n,k}$ *for all* $n \geq 3$, $2 \leq k \leq n-1$

5.4 Two questions about bracketing sequences

Recall that we consider the sorting tree $T(n) = T(\iota_n)$ to have labels on its edges, so that the edge from the child $\pi \in S_n$ to its parent $\Pi(\pi)$ is labelled by the bracketing sequence $b \in B_n$ associated to π by the sorting operation. It may help to think of this bracketing sequence as being associated to π by a function $b : S_n \to B_n$, so that $b(\pi)$ is really associated to the node labelled by π rather than to the edge π—$\Pi(\pi)$.

An alternative way to think about the bracketing sequence $b(\pi)$ is to think of the permutation which must be applied to π to obtain $\Pi(\pi)$. If $\Pi(\pi) = \sigma(\pi)$, then there exists a unique permutation ρ on the first level of $T(n)$ such that $\sigma(\rho) = \Pi(\rho) = \iota_n$. If ρ is this permutation, then clearly $\sigma = \rho^{-1}$. One has the option of thinking about the bracketing sequence b or, equivalently, about the permutation σ, but on the whole the interpretation in terms of bracketing sequences seems to cause less confusion. One observation which *is* facilitated by considering the permutation corresponding to a bracketing sequence is the following.

Proposition 5.4.1 *If $\Pi(\pi) = \Pi(\rho)$ but $\pi \neq \rho$, then $b(\pi) \neq b(\rho)$.*

Proof: Since $\pi \neq \rho$, a different permutation must be applied to each to obtain $\Pi(\pi)$. These different permutations correspond to different bracketing sequences. □

The proposition states that siblings in the tree $T(n)$ are associated to different bracketing sequences. The root node, ι_n, has one child for *each* of the $c_n - 1$ bracketing sequences *other* than one corresponding to the identity permutation, namely ()()()...(). Any other node, π, has $f(\pi)$ children, and these correspond to $f(\pi)$ different bracketing sequences, forming a subset of B_n. Instead of calculating $f(\pi)$ by characterizing the children of π, therefore, we could characterize the bracketing sequences which label the children.

The following questions about bracketing sequences and the tree $T(n)$ were suggested by Lauren Rose.

Question 5.4.2 *For a given $b^* \in B_n$, how many times does the bracketing sequence b^* appear as an edge-label in the tree $T(n)$? That is, for how many $\pi \in S_n$ is b^* associated to π by the sorting algorithm?*

Question 5.4.3 *For a given $b^* \in B_n$, how often does b^* appear as an label on the kth level of $T(n)$? That is, for how many $\pi \in \Delta kSS_n$ is b^* associated to π by the sorting algorithm?*

An alternate formulation for the second question would ask "for a given $\sigma \in S_n$, for how many $\pi \in \Delta k SS_n$ is $\Pi(\pi) = \sigma^{-1}(\pi)$?" Another alternate formulation would ask "for a given $b \in B_n$, for how many $\pi \in \Delta(k-1)SS_n$ does b label one of the children of π?" Our approach, which will answer the first question for all b^* and the second for some, will make strong use of the structure of the bracketing sequences. We first provide a recursive answer to Rose's first question.

Definition 5.4.4 *For $b \in B_n$, let $Z(b)$ be the number of $\pi \in S_n$ such that π is associated to b by the sorting algorithm. If $b^\emptyset \in B_n$ is the void bracketing sequence then $Z(b^\emptyset) = 1$.*

Theorem 5.4.5 *For any $b \in B_n$, if $b = b_L(b_R)$ where $b_L \in B_m$ and $b_R \in B_{n-1-m}$, then $Z(b) = \binom{n-1}{m} \cdot Z(b_L)Z(b_R)$.*

Proof: If $\pi \in S_n$ is associated with $b \in B_n$ by the sorting algorithm, and b is written in the form $b_L(b_R)$, the final closed parenthesis corresponds to the removal of the largest element, n, from the stack. Therefore its mate, namely the open parenthesis separating the two substrings b_L and b_R marks the point where n was placed on the stack.

If $b_L \in B_m$, this locates n at the $(m+1)$th position of π. We can write π as $\pi_L n \pi_R$, where $\pi_L \in S_m$. When n is processed, it clears the stack. Therefore, none of the elements occupying the first m positions will interact with any of the elements in the final $n-1-m$ positions. We are free to choose these elements, in any of $\binom{n-1}{m}$ ways.

Once the elements to fill the first m positions have been chosen, they are to be formed into the permutation sequence π_L. Clearly, π_L must be associated with the bracketing sequence b_L. But we know recursively how many permutations of length m are associated with this bracketing sequence, namely $Z(b_L)$. This also counts the number of permutation sequences on a given set of m elements which are associated with b_L.

The formation of the remainining $n-1-m$ elements into a permutation sequence associated with the bracketing sequence $b \in B_{n-1-m}$ can be performed *independently* in $Z(b_R)$ ways, in light of the fact that none of the elements of π_L interacts with any element of π_R.

After making the initial division of elements into a left and a right set and then independently forming the left and right subsequences, we have counted *all* the permutations associated with b. □

This essentially settles Rose's first question in a satisfactory manner. The second question is considerably more involved, and will be taken up in the following section.

Note that for a given $b^* \in B_n$, the unique permutation $\rho \in 1SS(n)$ such that $b(\rho) = b^*$ is the lexicographically least of all the permutations associated to b^*. Suppose we construct all the permutations associated to b^* in the manner of the above proof. The lexicographically least will be obtained if, whenever we have a choice, we always elect to place the smallest available elements on the left and the largest on the right. The permutation we determine will then have no subsequence of type 231. But this is exactly the property of ρ.

5.5 The spectrum of a bracketing sequence

In this section we begin our investigation of the second question of Lauren Rose. Given a bracketing sequence $b \in B_n$, how many times does b appear as a label on each level of the sorting tree $T(n)$? To further discussion of this problem, we make the following definition.

Definition 5.5.1 *The* spectrum *of a bracketing sequence* $\mathcal{S}[b] = (s_0, s_1, s_2, \ldots, s_{n-1})$ *records the number of times b appears as a label on each level of the tree $T(n)$. That is, s_k is the number of permutations $\pi \in \Delta kSS_n$ associated to b.*

Note that the spectrum is defined relative to a given permutation length n. So we should more properly write $\mathcal{S}_n$, but we omit the subscript for clarity. This should not cause confusion.

Let $b^= = ()()\cdots()$. We observe immediately that $b^=$ labels only the identity permutation ι_n, so that $\mathcal{S}[b^=] = (1, 0, \ldots, 0)$. On the other hand, if $b \neq b^=$, we know that b labels a unique permutation on the first level of $T(n)$, so that if $\mathcal{S}[b] = (s_0, s_1, \ldots, s_{n-1})$ then $s_0 = 0$ and $s_1 = 1$. Of course, the sum of the terms of the spectrum, $\sum_{k=0}^{n-1} s_k = Z(b)$, where Z is the function of definition 5.4.4.

Our first result will characterize recursively those bracketing sequences which have spectrum $\mathcal{S}[b] = (0, 1, 0, \ldots, 0)$ or $\mathcal{S}[b] = (1, 0, 0, \ldots, 0)$. Since bracketing sequences always begin $(0, 1, \ldots)$ or $(1, 0, \ldots)$, the bracketing sequence b will have one of these spectra precisely when $Z(b) = 1$.

Theorem 5.5.2 *If $b \in B_n$, $Z(b) = 1$ if and only if b can be written in either of the forms (b') or $b'()$ where $Z(b') = 1$.*

Proof: By theorem 5.4.5, if $b = b_L(b_R)$ then $Z(b) = \begin{pmatrix} n-1 \\ m \end{pmatrix} \cdot Z(b_L)Z(b_R)$. The product $Z(b)$ will be equal to 1 precisely if each term of the product is equal to 1. The binomial coefficient $\binom{n-1}{m}$ will be 1 only if either b_L or b_R is void. Therefore b must be of either of the forms $b_L()$ where $Z(b_L) = 1$; or (b_R) where $Z(b_R) = 1$. □

Corollary 5.5.3 *The number of $b \in B_n$ with $Z(b) = 1$ is 2^n.*

Proof: The proof is by induction on n. For $n = 1$, the unique bracketing sequence () has spectrum $\mathcal{S}[()] = (1)$. By the preceding theorem, each bracketing sequence in $b \in B_{n-1}$ with $Z(b) = 1$ can be extended to two such sequences in B_n. Since no well-formed sequence of the form $b()$ is also of the form (b') for nonvoid b, none of these sequences are the same. □

Example 5.5.4 *We may now refer to the function Z as the* spectrum sum *when it seems appropriate to do so. It may help to refer to the sorting tree for n = 3 pictured in chapter 4. Of the $c_3 = 5$ bracketing sequences of length 3, the number with spectrum sum 1 is $2^{3-1} = 4$. Their spectra are*

$$\begin{aligned}\mathcal{S}[()()()] &= (1,0,0)\\ \mathcal{S}[(()())] &= (0,1,0)\\ \mathcal{S}[(())()] &= (0,1,0)\\ \mathcal{S}[((()))] &= (0,1,0)\end{aligned}$$

As always, the special sequence $b^= = ()()()$ has spectrum $\mathcal{S}[b^=] = (1,0,\ldots,0)$. All other spectra begin $(0,1,\ldots)$, including the spectrum of the fifth and final bracketing sequence:

$$\mathcal{S}[()(())] = (0,1,1)$$

Since there is always exactly one sequence $b^= = ()()\cdots()$ with spectrum $(1,0,\ldots,0)$, the corollary states that the number of sequences $b \in B_n$ with $Z(b) = (0,1,0,\ldots,0)$ is $2^{n-1} - 1$.

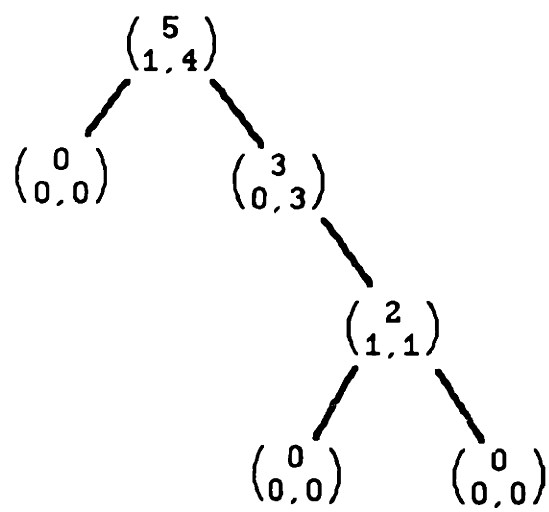
Definition 5.5.5 *If $b \in B_n$ is a bracketing sequence, the* choose-tree *associated with b, $\tilde{T}(b)$ is a rooted binary tree, with nodes labelled with binomial coëfficients, defined recursively as follows. For the unique sequence $b^1 \in B_1$, namely the sequence '()', $\tilde{T}(b^1)$ is a single node labelled by $\binom{0}{0,0}$. Otherwise, if $b = b_L(b_R)$ where $b_L \in B_m$ and $b_R \in B_{n-1-m}$, then the root of $\tilde{T}$ is labelled by $\binom{n-1}{m, n-1-m}$, the left subtree of the root is $\tilde{T}(b_L)$, and the right subtree of the root is $\tilde{T}(b_R)$.*

It follows immediately from the definition that if $b \in B_n$ then $\tilde{\mathcal{T}}(b)$ has n nodes. It should also be clear that the bracketing sequence b can be reconstructed from $\tilde{\mathcal{T}}(b)$, so that $\tilde{\mathcal{T}}(b_1) = \tilde{\mathcal{T}}(b_2) \Rightarrow b_1 = b_2$. The choose-tree $\tilde{\mathcal{T}}$ amounts to a visual record of the application of theorem 5.4.5 to calculate the spectrum sum $Z(b)$. Indeed, $Z(b)$ is just the product over the entire tree $\tilde{\mathcal{T}}$ of the values of the labelling binomial coëfficients. Note, however, that the labels are the actual symbols of the binomial coëfficients, and not their values. We have included two arguments on the bottom of each coëfficient in order to stress the symmetry of the situation, and to recall which is associated with the left subtree, and which with the right. Note also that all leaves of the choose-tree are labelled $\binom{0}{0,0}$.

Example 5.5.6 *The permutation* $(3,6,5,2,4,1)$ *corresponds to the following bracketing sequence*

$$\begin{array}{cccccccccccc} & 3 & & & & 2 & & & 1 & 4 & 5 & 6 \\ (&) & (& (& (&) & (& (&) &) &) &) \\ 3 & & 6 & 5 & 2 & & 4 & 1 & & & & \end{array}$$

The choose tree associated to this bracketing sequence is



If we were actually to calculate the value of the symbols at each node, we would find that any binomial coëfficient with a 0 as one of its bottom arguments contributed only a trivial factor of 1 to the product $Z(b)$. The the choose-symbols $\left(\begin{array}{c} n-1 \\ n-1,0 \end{array}\right)$ and $\left(\begin{array}{c} n-1 \\ 0,n-1 \end{array}\right)$ do not offer any practical 'choice' at all. We may wish to leave them out of our tree, as in the following definition.

Definition 5.5.7 *The reduced tree of $b \in B_n$, $\mathcal{T}(b)$ is obtained from the choose-tree $\tilde{\mathcal{T}}(b)$ by suppressing all nodes whose labelling binomial coëfficient has value 1. Thus, all leaves are deleted. Any remaining node, x, which is labelled by a symbol of value 1 has at most one nonempty subtree. Thus if x is the left (right) subtree of y, x can be deleted, and its child (if any) made to replace x as the left (right) child of y.*

Example 5.5.8 *The bracketing sequence $b = ()((()(())))$ of the previous example has the following reduced tree $\mathcal{T}(b)$*

$$\begin{tikzcd} \binom{5}{1,4} \arrow[dr] & \\ & \binom{2}{1,1} \end{tikzcd}$$

We can see from the reduced tree that there are $\left(\begin{array}{c} 5 \\ 1 \end{array}\right) \cdot \left(\begin{array}{c} 2 \\ 1 \end{array}\right) = 10$ permutations in S_6 corresponding to this bracketing sequence; that is that $Z(b) = 10$. These permutations are

165243	165342
265143	265341
365142	365241
465132	465231
564132	564231

Notice that in constructing these permutations from the bracketing sequence $b = ()((()(())))$, the elements which are entered when the nontrivial nodes of the reduced tree are processed, in this example those elements falling in positions 2 and 5 of the permutations, are exactly the *peaks* of the permutation. That is, they are those elements which fall immediately between two smaller elements.

Example 5.5.9 *If $b \in B_n$ is one of the 2^{n-1} bracketing sequences of length n with $Z(b) = 1$, every node in the choose-tree $\tilde{\mathcal{T}}(b)$ has a label of value 1. Consequently, such a b will have an empty reduced choose-tree $\mathcal{T}(b)$*

This example demonstrates the loss of information involved in considering only the reduced tree. Whereas the unique bracketing sequence associated with a full sorting tree can easily be determined, there may be many bracketing sequences associated to a given reduced tree. For instance, the choose-tree with no nodes is associated with 2^{n-1} bracketing sequences. We give the general result in the following theorem, after first defining a symbol which counts the number of bracketing sequences of length n associated to a given reduced tree.

Definition 5.5.10 *For a given $n \in Z^+$, if $\mathcal{T}^*$ is a reduced choose-tree whose root has the label $\binom{m}{m-k, k}$ for some $m < n$, then*

$$W_n(\mathcal{T}^*) = |\{b \in B_n : \mathcal{T}(b) = \mathcal{T}^*\}|$$

Theorem 5.5.11 *For a given $n \in Z^+$, if $\mathcal{T}^*$ is a reduced choose-tree whose root has the label $\binom{m}{m-k, k}$ for some $m < n$ and if $\mathcal{T}^*$ has v nodes, then*

$$W_n(\mathcal{T}^*) = 2^{n-1-2v}$$

Proof: We wish to determine all of the bracketing sequences associated with a given reduced tree. Perhaps the easiest way to think about this result is to consider a canonical permutation associated to each bracketing sequence. A convenient one is the lexicographically least, which is, as we observed at the end of section 5.4, also the unique permutation which is completely sorted by the bracketing sequence.

If the given reduced tree $\mathcal{T}^*$ has no nodes, then it corresponds to 2^{n-1} bracketing sequences, as we saw above. Each of these bracketing sequences is associated with only the canonical permutation. These 2^{n-1} permutations are built up by inserting the n elements into the permutation in decreasing order, beginning with n, and never inserting an element in such a way that the remaining elements are divided in a nontrivial manner. Thus, n is placed in either the first position [corresponding to a pair of brackets on the outside of the bracketing sequence, $b^* = (b_R)$] or in the last position [corresponding to pair of brackets at the end of the bracketing sequence, $b^* = b_L()$]. The element $n - 1$ is placed in either the leftmost or rightmost of the remaining positions, and so forth. This procedure corresponds to $n-1$ independent choices between leftmost and rightmost positions. (There are only $n-1$ choices, rather than n, because there is only one position remaining to be occupied by the smallest element, 1.)

To this point in the proof, we have said nothing new, yet we have suggested the form of proof of the general result. If the root node in the reduced tree $\mathcal{T}^*$ is labelled $\binom{m}{m-k, k}$ with $m = n-1$, then the position occupied by the element n is determined. (It is position $m-k+1$.) Otherwise, n might be located in either the rightmost or leftmost position, as neither offers a nontrivial choice. We continue to insert elements up to element $m+1$, each time making an either-or choice. When we insert element $m+1$, we process a node from the reduced tree, but lose two choices. One is lost because the position of element $m+1$ is fixed, and not free. The other is lost because, rather than having one element whose position will be forced at the end, we now have two, one on the right and the other on the left. Thus our total number of options is reduced by a factor of 2^{2v} if there are v nontrivial choices to be made. □

Example 5.5.12 *Consider the tree of example 5.5.8:*

$$\begin{array}{ccc} \binom{5}{1,4} & & \\ & \searrow & \\ & & \binom{2}{1,1} \end{array}$$

This tree is associated with $2^{6-1-2\cdot 2} = 2$ *bracketing sequences of length* 6. *Each of these bracketing sequences is associated with* 10 *permutations. We list the* 2 *bracketing sequences, and the lexicographically least permutation associated with each, below.*

()((()(())))	165243
()(()(())())	162435

Example 5.5.9 may also prepare us somewhat for the following theorem, the major result of this section.

Theorem 5.5.13 *The spectrum of a bracketing sequence is an invariant of the reduced tree associated to the sequence. In other words, if $b, b' \in B_n$ are such that $\mathcal{T}(b) = \mathcal{T}(b')$, then $\mathcal{S}(b) = \mathcal{S}(b')$*

Proof: Suppose that $\mathcal{T}(b) = \mathcal{T}(b')$, so that in particular $Z(b) = Z(b')$. The $Z(b)$ permutations associated to b and the $Z(b)$ permutations associated to b' can be put into one-to-one correspondence in an obvious manner: every a nontrivial choice is made on the one hand, make the same choice on the other.

More formally, given a permutation π associated to b, we obtain a permutation π' associated to b', as follows. Suppose the root of $\mathcal{T}(b)$ is labelled $\begin{pmatrix} m-1 \\ m-1-k, k \end{pmatrix}$. Then the elements $n, n-1, \ldots, m+1$ must be entered into both permutations in a fixed manner, determined by the respective bracketing sequences. The element m is also entered in a prescribed position, and the $m-1$ elements $1, 2, \ldots, m-1$ are in each case divided into a set of $m-k-1$ elements on the left and k on the right. Choose the same $m-k-1$ of these elements to precede the element m in π' as precede m in π. Procede recursively to fill the space of length $m-k-1$ on the left according to the left subtree of the root, and the space of length k on the right according to the right subtree.

We claim that the permutation π' obtained from π in this fashion is on the same level of the sorting tree $T(n)$. More strongly, we claim that $\Pi(\pi) = \Pi(\pi')$. The proof (which will actually be for general permutation sequences, rather than for permutations) will be by induction on the length of π.

Let π be a permutation sequence of length n associated with the bracketing sequence b, and let π' associated with b' be obtained according to the above procedure. (In particular, π' is built up out of the same set of elements as π). If π has length 1, it is clear that $\pi' = \pi$.

Otherwise, π may be written in the form $\pi = \rho_L \alpha m \beta \rho_R$, where the elements of ρ_L and of ρ_R are each greater than m, and the elements of α and β are each smaller than m. The elements of ρ_L and of ρ_R have been entered in decreasing order, beginning at the extremities of the permutation and working inwards. Observe that $\Pi(\pi) = \Pi(\alpha)\Pi(\beta)m, m+1, \ldots, n$.

Then if π' is obtained from π according to the procedure above, write $\pi' = \rho'_L \alpha' m \beta' \rho'_R$. Because of the nature of the procedure by which π' is obtained, α' consists of the same choice of elements as α. Also, because α' is obtained from α by the recursive application of the same procedure, the induction hypothesis states that $\Pi(\alpha) = \Pi(\alpha')$. Similarly, $\Pi(\beta) = \Pi(\beta')$. It only remains to compare the two forms

$$\begin{aligned} \Pi(\pi) &= \Pi(\alpha)\Pi(\beta)m, m+1, \ldots, n \\ \Pi(\pi') &= \Pi(\alpha')\Pi(\beta')m, m+1, \ldots, n \end{aligned}$$

to conclude inductively that $\Pi(\pi) = \Pi(\pi')$. □

This theorem allows us to write the spectrum of a bracketing sequence as a function of its associated reduced tree. Thus if $\mathcal{T}^* = \mathcal{T}[b]$, we may write $\mathcal{S}[\mathcal{T}^*] = \mathcal{S}[b]$.

Example 5.5.14 *The two bracketing sequences of the previous examples,* ()((()(()))) *and* ()(()(())()), *have the same reduced tree and consequently the same spectrum. In fact,* $\mathcal{S}[()((()(())))] = \mathcal{S}[()(()(())())] = (0,1,8,1,0,0)$, *which can be verified by writing the* 10 *permutations corresponding to each bracketing sequence, and applying the sorting algorithm.*

We thus have a convenient formula for the number of bracketing sequences associated with a given reduced tree, $W_n(\mathcal{T})$, and understand that each of these has the same spectrum. The next natural step would be to determine a way to read off these spectra from the reduced tree. Unfortunately, this still seems to be difficult. In the following section we will have some success in determining the third term of the spectrum, the one counting the number of two-stack sortable permutations, from the reduced tree. For now, we leave one special case in which the spectrum can be read off directly.

We already know that if $\mathcal{T}[b]$ is the empty tree, which we can call $\mathcal{T}^0$, then $\mathcal{S}[b] = (0,1,0,\ldots,0)$, except for the special case $\mathcal{S}[b^=] = (1,0,0,\ldots,0)$. [This is the one exception to the above theorem about spectral invariance, which we finessed in the proof.] For this reason, let us agree that $\mathcal{S}[\mathcal{T}^0] = (0,1,0,\ldots,0)$.

We turn from the empty tree to a tree with a single node.

Theorem 5.5.15 *If* $\mathcal{T}^*$ *is the reduced tree consisting of the single node* $\begin{pmatrix} m \\ m-1-k, k \end{pmatrix}$ *then*

$$\mathcal{S}[\mathcal{T}^*] = \left(0, \underbrace{\begin{pmatrix} k-1 \\ k-1 \end{pmatrix}, \begin{pmatrix} k \\ k-1 \end{pmatrix}, \begin{pmatrix} k+1 \\ k-1 \end{pmatrix}, \ldots, \begin{pmatrix} m-1 \\ k-1 \end{pmatrix}}_{m-k+1}, \underbrace{0,\ldots,0}_{n-m}\right)$$

Proof: The proof relies on theorems 4.2.10 and 4.2.14, which give conditions locating a permutation $\pi \in S_n$ within the sorting tree $T(n)$. First note that (in the language of these theorems) if $\mathcal{T}[b(\pi)] = \mathcal{T}^*$, then every forbidden pattern, of any order, in π is pure. This follows from the observation following example 5.5.8 that the elements in the permutation corresponding to the nodes of the reduced tree are exactly the peaks of the permutation. Since $\mathcal{T}^*$ has only one node, there is only one peak in the interior of π. In the definition of a pure forbidden pattern, 4.2.13, the pattern of increases and decreases in the subsequence (b_1, x, b_2, c, a) ensures that an impure pattern (i.e., one containing such a subsequence) must have at least twin peaks.

If the highest order forbidden pattern in our permutation π is of order r, then theorem 4.2.10 asserts that π appears on or above level $r+1$ of the sorting tree $T(n)$. On the other hand, since our the forbidden patterns in our permutation π are all pure, theorem 4.2.14 asserts that π appears below level r. Therefore, $\pi \in \Delta(r+1)SS_n$. It remains to find the highest order forbidden pattern in π, for each of the $Z(b)$ permutations associated to any of the bracketing sequences b having reduced tree $\mathcal{T}^*$.

As such a π has only one peak, namely the element $m+1$, there will be a maximal order forbidden pattern having $c = m+1$. The order of this pattern will be determined by the smallest element which is chosen to follow the element $m+1$ in π. Consider the selection of k of the elements $\{1, 2, \ldots, m\}$ to fall to the right of element $m+1$. If the least of these is $m-(k-1)-s$ for some $s \geq 0$, then of the $k-1+s$ greater elements, $k-1$ will also follow the element $m+1$ but s will precede this element. This set of s elements, together with the pair $m+1 > m-k-s$, will be the maximal order forbidden pattern in π, a pattern of order s. There are $\binom{k-1+s}{k-1}$ ways to choose the $k-1$ elements larger than $m-(k-1)-s$ which are also to follow the element $m+1$.

Hence, the number of π associated with b which are exactly $s+1$-stack sortable is $\binom{(k-1)+s}{k-1}$, for all $s \geq 0$. This is precisely the statement of the theorem. □

Note that each of the components of the spectrum $\mathcal{S}[\mathcal{T}^*]$ given in the above theorem depends only on the variable k. The other variable appearing in the reduced tree $\mathcal{T}^*$, namely m, only determines the number of nonzero terms appearing in the spectrum.

5.6 What the spectra say about level 2

In the previous section, we argued that the spectrum of a bracketing sequence was an invariant of the associated reduced tree. We next prove a stronger statement about the first nontrivial coëfficient of the spectrum, namely the one which counts the number of exactly 2-stack sortable permutations associated with a given bracketing sequence. Let us use $t_r[b]$ to denote the number of permutations $\pi \in rSS_n$ associated to a bracketing sequence b, so that if $\mathcal{S}[b] = (s_0, s_1, s_2, \ldots, s_{n-1})$, then $t_r[b] = \sum_{i=1}^{r} s_i$. In particular, $t_2[b] = s_0 + s_1 + s_2 = s_2 + 1$.

Also recall that the reduced tree of a bracketing sequence b is a rooted binary tree, with each node x labelled with a binomial coëfficient, $\left(\begin{array}{c} m_x \\ m_x - k_x, k_x \end{array} \right)$. From the reduced tree $\mathcal{T}[b]$ we define the k-tree $\mathcal{T}^k[b]$ to be an isomorphic tree with the node corresponding to x labelled by k_x. We prove the following result.

Theorem 5.6.1 *Given a bracketing sequence $b \in B_n$, the number of 2-stack sortable permutations in S_n associated to b is an invariant of the k-tree $\mathcal{T}^k[b]$.*

Proof: We use the characterization of theorem 4.2.18. That is, a permutation π is two-stack sortable unless it contains a subsequence of type 2341, or a subsequence of type 3241 which is not part of a subsequence of type 35241.

When a permutation is created in accordance with bracketing sequence b having reduced tree $\mathcal{T}[b]$, then the elements which are peaks in the permutation are precisely those which are entered when a node of the reduced tree is processed. Clearly, if any element serves as the largest element of a forbidden subsequence of type 2341 or 3241, then a peak does so. We enter elements in such a way as to ensure that they do not.

Processing a node x of the reduced tree labelled $\left(\begin{array}{c} m_x \\ m_x - k_x, k_x \end{array} \right)$ corresponds to selecting, among the elements to be filled into a contiguous region of the permutation, those k_x which are to fall to the right of, and those $m_x - k_x$ which are to fall to the left of, the largest element. If any two of those selected to fall to the left are larger than the least of those selected to fall to the right, then a forbidden subsequence of type 2341 or 3241 will necessarily be introduced. (f it were to turn out eventually to be of type 3241, then it would not be part of a subsequence of type 35241, as the elements on the right are to be entered contiguously, and all are smaller than the largest of the m_x elements.) Hence, to avoid introducing one of these forbidden subsequences, the k_x elements chose to fall on the right must be selected from among the largest $k_x + 1$ elements available. This number is independent of the number of elements falling to the left.

We next check that the selection of k_x of these $k_x + 1$ elements to fall on the right (or equivalently, of one to fall on the left) imposes the same restrictions on processing the subtrees of x, regardless of the number m_x. It certainly makes no difference to the way the elements can be entered in the substring on the left; so long as these are entered in such a way as to avoid the forbidden subsequences they cannot interact negatively with any elements outside this substring.

On the right, we must proceed with more care, as it is possible to introduce a forbidden subsequence of type 2341 in the following manner. A node x is processed, meaning that a large element, say c_1, is entered at a fixed spot and some elements are selected to be placed to its left, some to the right. At most one of the elements on the left, say b_1, is permitted to be larger than any of the elements on the right. Say the minimum element on the right is a_1. Later, we process a node y in the right subtree of x. A large element, c_2 is fixed, and of the elements then placed to the left of c_2, one b_2 is permitted to be larger than the minimum element on the right, a_2. Since, alone among the elements to the left of and smaller than c_1, b_1 may be larger than some elements to the right of c_1, it might well be the case that b_1 is larger than either or both of the elements b_2 and a_2, but smaller than c_2. If it is larger than a_2 but smaller than b_2, then (b_1, b_2, c_2, a_2) is a forbidden subsequence of type 2341. If, however, b_1 is larger than b_2, but smaller than c_2, then (b_1, b_2, c_2, a_2) is of type 3241 but is part of the subsequence $(b_1, c_1, b_2, c_2, a_2)$ and so not forbidden by the characterization of $2SS_n$.

None of this, however, makes any use of the size of the set on the left to which the elements b_1 and b_2 belong. Knowing how many elements to the right of c_i are smaller than b_i is sufficient. □

The form of this proof suggests a technique for gaining information about the number of two-stack sortable permutations. A starting point would be to find a way to determine the number $t_2[b]$ as a function of the k-tree $\mathcal{T}^k[b]$. Experimentation suggests that to each rooted binary tree T there is associated a polynomial function of the nodes, and that $s_2[b]$, for each b associated with a k-tree $\mathcal{T}^k[b]$ of shape T, can be found by evaluating this polynomial at the specific values of the labels on $\mathcal{T}^k$. These functions appear to be products of binomial coëfficients with integer coëfficients $+1$ and -1, suggesting inclusion-exclusion formulæ. Furthermore, the polynomials are homogeneous products of binomial coëfficients in that the sum over the lower terms in the binomial coëfficients of each monomial is constant, and equal to the depth of the binary tree.

Although it is not difficult to compile a table of such formulæ, we shall not attempt to do so here, as it does not seem useful to accumulate such results in an *ad hoc* fashion without developing a general framework. We will content ourselves with one observation, based directly on the above proof of theorem 5.6.1. If $\mathcal{T}^R$ is a k-tree whose root has an empty left subtree, and $\mathcal{T}^\bullet$ is another k-tree, let $\mathcal{T}^\times$ be the k-tree formed by making $\mathcal{T}^\bullet$ the left subtree of the root of $\mathcal{T}^R$. Then if $\mathcal{T}^R = \mathcal{T}^k[b_1]$, and $\mathcal{T}^\bullet = \mathcal{T}^k[b_2]$, and $\mathcal{T}^\times = \mathcal{T}^k[b_\times]$, then $t_2[b_\times] = t_2[b_1] \times t_2[b_2]$. This is a direct result of the observation in the proof of theorem 5.6.1 that the decision made at the root does not affect any of the decisions made while processing the left subtree.

If we could develop enough similar results to be able to read off the value of $t_2[b]$ from any k-tree $\mathcal{T}^k[b]$, we might be able to find a clever way to sum $t_2[b]$ over all reduced trees, using theorem 5.5.11 on the number of bracketing sequences associated to a given reduced tree, to find the total value $|2SS_n|$. Alternatively, we might at least find a computationally efficient way to calculate $|2SS_n|$, and so establish whether there is a small counterexample to conjecture 4.2.19.

In the following subsection, we suggest other approaches to proving the conjecture of 4.2.19.

5.7 Other attempts to prove the level 2 conjecture

5.7.1 Appeal to forbidden subsequences

We have a characterization of the two-stack sortable permutations in terms of forbidden subsequences, which was given in theorem 4.2.18. We could try to use any of the techniques of chapters 2 and 3 to enumerate $2SS_n$.

In [18], Simion and Schmidt have had considerable success in enumerating intersections of the form $\cap_{i=1}^{r} S_n(\tau_i)$ where $\tau_i \in S_3$. Unfortunately, we have not quite written $2SS_n$ in the form $S_n(\tau) \cap S_n(\rho)$ for $\tau, \rho \in S_4$, because of the unusual condition that subsequences of type 3241 are only excluded if they do not form part of a subsequence of type 35241. Although this may not make the enumerative problem harder, and indeed the data for $2SS_n$ seem to fit a more attractive form than, for instance, $S_n(2341) \cap S_n(3241)$, it probably does make the problem harder from a forbidden subsequences approach. In any case, forbidden subsequence problems have proven to be difficult, and this is probably not the most promising approach.

5.7.2 A set of recurrence equations

One way, and perhaps the simplest way, in which we showed that $1SS_n$ was counted by the Catalan numbers was using the characterization $S_n(231)$, and showing, at the beginning of section 2.1, that these numbers satisfied the Catalan recurrence (1). We could try a similar approach to find a recurrence satisfied by the numbers $d_n = |2SS_n|$.

We will fix the position of the largest element, n, and sum over all possible positions. Suppose n appears in position j. Write a permutation $\pi \in 2SS_n$ in the form $\pi = \alpha n \beta$, where $\alpha \in S_{j-1}$ and $\beta \in S_{n-j}$. Then $\Pi(\pi) = \Pi(\alpha)\Pi(\beta)n$. In order for the permutation π to be two-stack sortable, we want this once-sorted permutation $\Pi(\pi)$ to be one-stack sortable. That is, we want $\Pi(\pi) \in S_n(231)$. This means, among other things, that $\Pi(\alpha) \in S_{j-1}(231)$ and $\Pi(\beta) \in S_{n-j}(231)$, so that $\alpha \in 2SS_{j-1}$ and $\beta \in 2SS_{n-j}$. How else could a forbidden subsequence of type 231 appear in $\Pi(\pi)$? We have dealt with the cases where all three terms of the forbidden subsequence appear in $\Pi(\alpha)$, and where all three terms appear in $\Pi(\beta)$. One remaining case has the first two terms in $\Pi(\alpha)$ and the third in $\Pi(\beta)$; the other has the first term in $\Pi(\alpha)$ and the second and third in $\Pi(\beta)$.

We deal first with the first of these two cases. Since $\Pi(\alpha)$ is a sorted permutation, its largest element appears on the right, and so its largest two elements are in increasing order. Therefore, if any two elements of α are larger than the smallest element of β, a forbidden subsequence of type 231 will appear under the first case. If at most one element of α is larger than the smallest element of β, no such forbidden subsequence will appear. This takes care of the first outstanding case. In addition to summing over the position of n, we will have a second nested sum, in which the large element of α is allowed to range. Otherwise, the elements of α are all the small elements, the elements of β all the large ones.

The other outstanding case is where the difficulty arises. How could a forbidden subsequence of type 231 appear with its first term in $\Pi(\alpha)$ and its other two terms in $\Pi(\beta)$? Since the second and third terms will be decreasing in a once-sorted permutation, a result of subsection 4.2.1 states that these terms must have appeared as the elements b and a of a wedge (b, x, a) in β. We need to count, then, not only the number of permutations $\beta \in 2SS_{n-j}$, but the number of these permutations which have no wedge (b, x, a) so that $b > y > a$ for a specified y, that value of y corresponding to the element selected to be the free-ranging maximum element of α. If the number of these is d^y_{n-j}, then the recurrence we seek is actually of the form

$$d_n = \sum_{j=1}^{n} \sum_{y=1}^{n-j} d_{j-1} \cdot d^y_{n-j}$$

. To complete this argument, we would have to find appropriate recurrences for the numbers d^y_{n-j}, presumably each involving numbers of a similar form, and solve the set of recurrences.

Note the similarity in these arguments to the work with k-trees in section 5.6. Perhaps we are repeating the same argument in a different guise, or perhaps there is an idea in one approach which has been overlooked but which might usefully contribute to the other.

5.7.3 Generalized bracketing sequences

Another way to think about the members of $1SS_n$ was to consider the bracketing sequence uniquely associated with each one-stack sortable permutation. In the literature, these sequences have frequently been written with the letters S and X representing pushes and pops from the stack, rather than the symbols '(' and ')'. We have preferred the latter, because they suggest naturally the characterization of a well-formed bracketing sequence in terms of matching pairs of parentheses. But let us consider generalizing the sequences of S's and X's, representing pushes and pops, to two stacks. We use sequences of S's, T's and X's, where an S represents placing an element on the first stack, a T represents a transfer between stacks, and an X represents removing an element from the final stack. We only need three letters, not four, because once an element p is removed from the first stack it might as well be placed directly on the second one. If we wished to remove some elements from the second stack first, we could have removed these before p was taken from the first stack.

We wish to count sequences in which each operation S, T and X is performed n times, one for each element of an input permutation. Note that two such sequences of letters are equivalent if an adjacent pair SX is replaced by XS. This is because if there is no intervening T, the operations of placing an input on the first stack and removing an output from the second can be performed equivalently in either order. So we might as well assume that between each consecutive pair of T's we have a string of S's followed by a string of X's. As a first step, we might locate all the T's, in $\binom{3n}{n}$ ways. This seems promising, as a factor of this form appears in our conjectured expression in 4.2.19.

But we need to restrict our attention not only to sequences which are well-formed in the sense that we never try to remove from either stack more elements than are actually there, but further to well-formed sequences which are inequivalent, in terms of producing the same effect when performed on an input permutation. This seems difficult, as not only are, for instance, the sequences $STXSTX$ and $STSXTX$ equivalent, but the sequences $STSTXX$ and $SSTXTX$ are as well.

5.7.4 A paper of Tutte's

The first 11 terms of the sequence of $|2SS_n|$ are 1, 2, 6, 22, 91, 408, 1938, 9614, 49 335, 260 130, 1 402 440. In our conjecture 4.2.19 we noted that these numbers fit the formula $\frac{2}{(n+1)(2n+1)}\binom{3n}{n}$. Taking this sequence to Sloane's book of integer sequences [19] turns up two references. The second of these is glancing, but the first, one of Tutte's "census" papers [22] derives at some length the formula $\frac{2}{(n+1)(2n+1)}\binom{3n}{n}$ for the number of nonseparable planar graphs counted by edges.

Tutte's argument, which is involved and relies on results from a previous paper, uses generating functions and Lagrange inversion. This seems to be a promising technique. Can the objects we wish to count, namely the two-stack sortable permutations, be shown to obey Tutte's generating functions, and his arguments reproduced in this new setting? Presumably, one of the approaches to the enumerative problem considered above would be used in accomplishing this. Alternatively, can the two-stack sortable permutations be placed in direct correspondence with Tutte's non-separable planar maps?

References

[1] ANDERS BJÖRNER AND MICHELLE L. WACHS, Permutation Statistics and Linear Extensions of Posets, preprint (1989)

[2] EUGENE CATALAN, Note sur une Équation aux différences finies, J. de Mathématiques pures et appliquées, 3 (1838), 508-516

[3] F. R. K. CHUNG, On Unimodal Subsequences, Journal of Combinatorial Theory, Series A 29 (1980), 267-279

[4] F. R. K. CHUNG, R. L. GRAHAM, V.E. HOGGATT, JR., M. KLEIMAN, The Number of Baxter Permutations, Journal of Combinatorial Theory, Series A 24 (1978), 382-394

[5] S. EVEN AND A. ITAI, Queues, Stacks and Graphs, Theory of Machines and Computations (1973)

[6] IRA GESSEL, Symmetric Functions and P-Recursiveness, Journal of Combinatorial Theory, Series A 53 (1990), 257-285

[7] D. E. KNUTH, "The art of computer programming, vol. 1," Addison-Wesley, Reading MA, 1973 (second edition), p. 239

[8] D. E. KNUTH, Permutations, matrices, and generalized Young tableaux, Pacific Journal of Mathematics 34 (1970), 709-727

[9] LAMÉ, Extrait d'une lettre de M. Lamé à M. Liouville sur cette question: Un polygone convexe étant donné, de combien de manieères peut-on le partager en triangles au moyen de diagonales?, J. de Mathématiques pures et appliquées, 3 (1838), 505-507

[10] ALAIN LASCOUX AND M.-P. SCHÜTZENBERGER, Schubert Polynomials and the Littlewood-Richardson Rule, Letters in Math. Phys. 10 (1985) 111-124

[11] P.A. MACMAHON, "Combinatory Analysis," Chelsea, 1960. Originally published by Camb. Univ. Press, London 1915/1916.

[12] SHAIY PILPEL, Descending Subsequences of Random Permutations, J.C.T. A 53 (1990), 96-116

[13] A. REGEV, Asymptotic values for degrees associated with strips of Young diagrams, Adv. in Math. 41 (1981), 115-136

[14] DANA RICHARDS, Ballot Sequences and Restricted Permutations, Ars Combinatoria 25 (1988), 83-86

[15] D. G. ROGERS, Ascending Sequences in Permutations, Discrete Mathematics 22 (1978), 35-40

[16] D. ROTEM, Stack sortable permutations, Discrete Mathematics 33 (1981), 185-196

[17] C. SCHENSTED, Longest increasing and decreasing subsequences, Canadian Journal of Mathematics 13 (1961), 179-191

[18] RODICA SIMION AND FRANK W. SCHMIDT, Restricted Permutations, European Journal of Combinatorics 6 (1985), 383-406

[19] N.J.A. SLOANE, "A Handbook of Integer Sequences," Academic Press (1973)

[20] KAREN STELLPFLUG, Discrete Representations of Semiorders, doctoral thesis, Dartmouth College 1990

[21] RICHARD P. STANLEY, On the Number of Reduced Decompositions of Elements of Coxeter Groups, European Journal of Combinatorics 5 (1984), 359-372

[22] W. T. TUTTE, A census of planar maps, Canadian Journal of Mathematics 15 (1963), 257