Chapter 33

Longest Increasing Subsequence

[This section was originally written by Anand Sarwate]

33.1 Introduction

In this paper we will investigate the connection between random matrices and finding the longest increasing subsequence of a permutation. We will introduce a model for the problem using a simple card game. Then we will talk about Young tableaux and their relation to the symmetric group. Representation theory and power-sum symmetric functions serve as the bridge between this combinatorial construction and random matrices. The presentation in this paper is largely modeled on that of Aldous and Diaconis [2], but is shorter and designed to be read by a wider audience. A reader with two semesters of abstract algebra and some probability should be able to follow the ideas presented here with little difficulty.

The problem in which we are interested in is as follows. Let π be a random permutation of the integers $1, 2, \ldots, n$. $\pi(i)$ is the i-th element of the permutation. Then an *increasing subsequence* $(i_1, i_2, \ldots, i_k)$ of π is a subsequence satisfying:

$$
\begin{array}{rcl}
i_1 < i_2 < & \ldots & < i_k \\
\pi(i_1) < \pi(i_2) < & \ldots & < \pi(i_k)
\end{array}
$$

We define $l(\pi)$ as the length of the *longest increasing subsequence*. We denote by L_n the integer valued random variable which takes on the value of $l(\pi)$. The purpose of this paper is to investigate the distribution of L_n and to express this distribution in terms of random matrices.

In the second section we discuss *patience sorting*, a simple card game model for computing $l(\pi)$ and provide some Monte Carlo simulations to show the empirical distribution of L_n. The third section deals with Young tableaux, a combinatorial construction with applications in representation theory and geometry. We describe these structures and prove the Schensted correspondence, which makes explicit the link between the symmetric group and the set of Young tableaux. The Schensted correspondence can also be used to compute the distribution of L_n. The fourth section provides a quick summary of some facts from representation theory and expresses the distribution of L_n in terms of characters of the irreducible representations of the symmetric group S_n. The fifth section discusses power-sum symmetric functions and their role in linking the symmetric and unitary groups. This allows us to express the distribution of L_n in terms of the matrix integral of the trace of a power of a unitary matrix. The last section covers some extensions of this work to other groups of matrices and other types of increasing subsequence problems.

33.2 Raking Leaves and Longest Increasing Subsequences

Suppose we're given a permutation $p = (p_1, \ldots, p_n)$ of the integers 1 through n. An *increasing subsequence* is a sequence of elements of p (not necessarily consecutive), $p_{i_1}, p_{i_2}, \ldots, p_{i_k}$, with $i_1 < i_2 < \cdots < i_k$ and $p_{i_1} < p_{i_2} < \cdots < p_{i_k}$. Our problem is to find an increasing subsequence that is as long as possible. There may be more than one; for example, $p = (3, 1, 4, 5, 9, 2, 6, 8, 7)$ has four longest increasing subsequences, including $(1, 4, 5, 6, 7)$ and $(3, 4, 5, 6, 8)$.

We can look at this as a problem on a directed acyclic graph, or dag. The vertices of the dag are the integers 1 through n, and there is an edge directed from i to j if $i < j$ and i occurs to the left of j in the permutation. An increasing subsequence is a directed path. The length of the longest increasing subsequence is the *height* of the dag.

Define a *leaf* as a vertex with no edges directed into it. The leaves in our example are 3 and 1. Appropriately for the season, we can find the height of the dag by an algorithm called *leaf raking*:

- repeat
- identify all the leaves
- delete all the leaves
- until no vertices remain

The number of iterations this loop needs to delete all the dag's vertices is its height. Consider the example. The first iteration deletes the leaves 3 and 1. The leaves of the remaining dag are 4 and 2. (Notice that leaves are the same as left-to-right minima.) After deleting 4 and 2, the only leaf is 5, and so on. Call the vertices deleted in a single iteration a *level*. We can write down a tableau showing the vertices by level, in the order they are deleted:

$$(31\|42\|5\|96\|87).$$

This tells us that longest increasing subsequences have length five, but how do we find one? Easy lemma: every vertex in the last level of the tableau (8 and 7 in the example) is the final vertex of some longest path. (Not every vertex in the first level need be the start of a longest path. Exercise: give an example.) Thus we can construct a longest path by starting at any vertex of the last level and working backwards: start with 8, say; the lemma promises us a vertex on the previous level with an edge to 8 (it's 6); the lemma promises an edge from 5 to 6; and an edge from either 4 or 2 to 5 (it's 4); and from either 3 or 1 to 4 (either will do, say 3). Reversing the order in which we chose the vertices gives the longest increasing subsequence $(3, 4, 5, 6, 8)$.

Leaf raking has an application to parallel solution of triangular linear systems. Suppose we wish to solve $Lx = b$ for x, given an n-by-n lower triangular matrix L and a right-hand side vector b. The directed graph of L^T has n vertices, with an edge directed from i to j if $i < j$ and $l_{ji} \neq 0$. (Note the transpose.) If L is nonsingular then its diagonal elements are nonzero, but we don't include loop edges for them. The graph is a dag because L is triangular. A leaf corresponds to a row of L (or column of L^T) whose only nonzero entry is on the diagonal. Thus all the entries of x that correspond to leaves can be computed independently and simultaneously. We can then rake up those leaves and proceed to the the next level of the dag. On an ideal parallel machine with arbitrarily many processors, we solve the whole system in a step per level of leaf raking.

33.3 Patience sorting: a model

Suppose we have a deck of cards numbered $1, 2, \ldots, n$ which we shuffle into a random order[1]. We turn up cards one at a time and place them into piles according to the following rule: *a lower number may be placed on top of a higher number or be placed into its own pile.* The goal is to end with as few piles as possible.

While this may not seem to be the most interesting of games at first glance, a simple strategy for playing yields an effective way of computing $l(\pi)$. The greedy strategy involves placing the current card on the leftmost pile possible. This game is best illustrated with an example. Say we have a deck of 10 cards in the following order:

$$8\ 3\ 7\ 9\ 2\ 5\ 4\ 1\ 10\ 6$$

The greedy strategy will play the game in the following manner:

$$\begin{array}[b]{c} 8 \end{array} \qquad \begin{array}[b]{c} 3 \\ 8 \end{array} \qquad \begin{array}[b]{cc} 3 & \\ 8 & 7 \end{array} \qquad \begin{array}[b]{ccc} 3 & & \\ 8 & 7 & 9 \end{array} \qquad \begin{array}[b]{ccc} 2 & & \\ 3 & & \\ 8 & 7 & 9 \end{array} \qquad \begin{array}[b]{ccc} 2 & & \\ 3 & 5 & \\ 8 & 7 & 9 \end{array}$$

$$\begin{array}[b]{ccc} 2 & 4 & \\ 3 & 5 & \\ 8 & 7 & 9 \end{array} \qquad \begin{array}[b]{ccc} 1 & & \\ 2 & 4 & \\ 3 & 5 & \\ 8 & 7 & 9 \end{array} \qquad \begin{array}[b]{cccc} 1 & & & \\ 2 & 4 & & \\ 3 & 5 & & \\ 8 & 7 & 9 & 10 \end{array} \qquad \begin{array}[b]{cccc} 1 & & & \\ 2 & 4 & & \\ 3 & 5 & 6 & \\ 8 & 7 & 9 & 10 \end{array}$$

It turns out that patience sorting is an easy way for us to not only calculate $l(\pi)$, but also to find an increasing subsequence in π of length $l(\pi)$.

Theorem 33.1. *Let π be the ordering of a deck of cards numbered $\{1, 2, \ldots, n\}$. Using the greedy strategy to play patience sorting using π will result in exactly $l(\pi)$ piles.*

Proof. We will first prove that the number of piles is at least $l(\pi)$ and then provide a construction to show that it is at most $l(\pi)$.

Let $i_1 < i_2 < \ldots < i_k$ be an increasing subsequence in π. Then $\pi(i_j)$ must always lie in a pile to the right of $\pi(i_{j+1})$ since we are only allowed to place cards on top of cards of smaller value. Therefore any valid strategy, and particularly the greedy strategy, will result in at least $l(\pi)$ piles.

We can use patience sorting to find an instance of an increasing subsequence, thereby showing that we have at most $l(\pi)$ piles. Every time we place a card c in a pile i that is not the first pile, draw an arrow from c to the top card d of the preceding pile $i - 1$. We know $d < c$ because in our greedy strategy, c would be placed on top of d if $d > c$. Note that each arrow goes from a later card to an earlier card. This is illustrated in Figure 33.1.

If we have k piles, call the the top card in the rightmost pile a_k. Then follow the arrow from a_k to a_{k-1} and so on. In this way we construct an increasing subsequence in π. Therefore we have at most $l(\pi)$ piles. $\square$

The reason the game is called patience *sorting* is because at the end, we can sort the cards into order. Card 1 will be at the top of some pile. Removing it, we are left with a patience-sorted deck

[1]We recommend the reader to make such a deck themselves to better follow along with the examples used in this section.

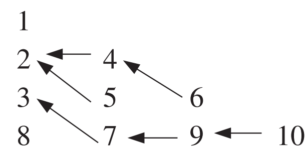


Figure 33.1: Constructing an increasing subsequence

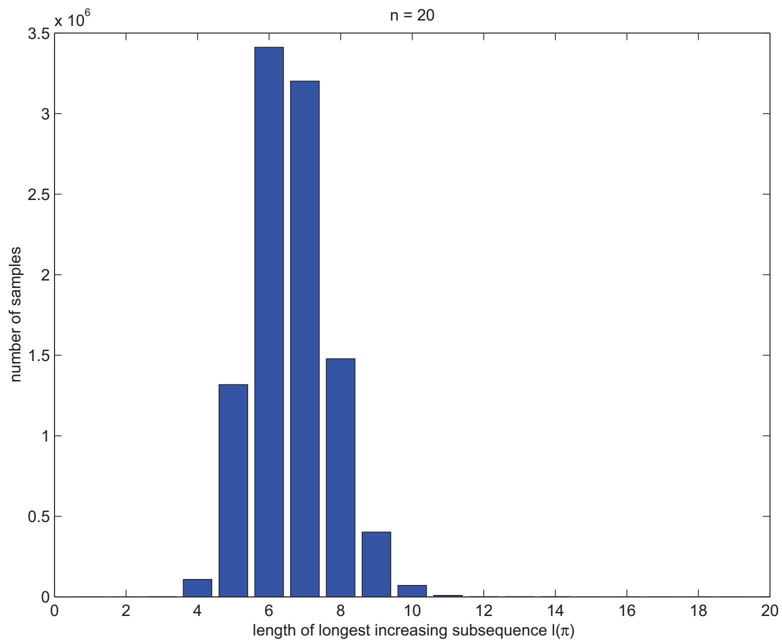


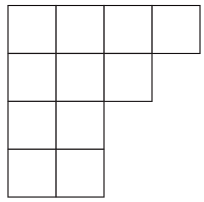
Figure 33.2: Histogram of number of piles for patience sorting on decks of 10 cards (left) and 20 cards (right) using 10^7 samples

of $\{2, 3, \ldots, n\}$, so card 2 must now be at the top of some pile. Proceeding as before we can sort the deck. As it turns out, however, this is also a convenient and easy way to compute $l(\pi)$.

In Appendix 33.8 we provide a MATLAB script to compute the distribution of L_n using this method. A histograms for $n = 10$ and $n = 20$ using 10^7 samples are shown in Figure 33.2. In the case of patience sorting, winning is not well defined, and instead we can arbitrarily choose a number of piles as a threshold for winning or losing. Readers more interested in patience sorting and its history should read the survey by Aldous and Diaconis [2], which also covers much of the other material in this paper.

33.4 Young Tableaux

We now turn to another way of looking at permutations using diagrams called *Young tableaux*. These are special structures which are used in invariant theory and group representations of the symmetric group S_n. They are also important in combinatorics and algebraic geometry, and in the theory of symmetric functions. What is most important for our problem is an important construction called the Schensted correspondence, which relates multiset permutations to ordered



1	2	7	10
3	5	9	
4	8		
6	11		

Figure 33.3: A Ferrers diagram for $(4, 3, 2, 2)$ and example of a Young tableau with shape λ

pairs of Young tableaux.

Let $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ be a set of integers such that $\sum \lambda_i = n$ and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Then we say λ is a *partition of* n and write $\lambda \vdash n$. We denote the number of entries in λ by $|\lambda|$, which in this case is equal to k. In the special case where $\lambda = (1, 1, \ldots, 1)$ we write $\lambda = 1^n$.

A *Ferrers diagram* with shape λ is a set of cells as shown in Figure 33.3, where row i has λ_i cells. A *standard Young tableau* is a Ferrers diagram with the numbers $1, 2, \ldots, n$ in the cells such each number is used once, and the entries increase along each row and down each column.

The *hook length* h_c of a cell c is the number of cells to the right of c in its row plus the number of cells below c in its column plus the cell c itself. Thus in the tableau in Figure 33.3 the hook length of the cell containing 5 is $h_c = 4$, and the hook length of the cell containing 8 is $h_c = 2$.

An interesting question is this: how many standard Young tableaux are there of shape λ? The answer is surprisingly simple, and given by the following theorem.

Proposition 33.2. *(Hook Formula) Let d_λ be the number of standard Young tableaux of shape λ. Then:*

$$d_\lambda = \frac{n!}{\prod_c h_c} \tag{33.1}$$

If we look at the tableau in Figure 33.3 we can calculate the number of tableau of that shape:

$$d_{(4,3,2,2)} = \frac{11!}{7 \cdot 6 \cdot 3 \cdot 1 \cdot 5 \cdot 4 \cdot 1 \cdot 3 \cdot 2 \cdot 2 \cdot 1} = 1320$$

This is far too many to explicitly verify, so we can look at an easier example. Consider $\lambda = (3, 2) \vdash 5$. The hook length formula tells us we can only construct $120/(4 \cdot 3 \cdot 1 \cdot 2 \cdot 1) = 5$ standard Young tableaux. These are shown in Figure 33.4. It is left as an exercise to the reader to prove there exist no more standard Young tableaux of that shape.

$$\begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 4 & 5 \\ \cline{1-2}\end{array} \quad \begin{array}{|c|c|c|}\hline 1 & 3 & 5 \\ \hline 2 & 4 \\ \cline{1-2}\end{array} \quad \begin{array}{|c|c|c|}\hline 1 & 2 & 5 \\ \hline 3 & 4 \\ \cline{1-2}\end{array} \quad \begin{array}{|c|c|c|}\hline 1 & 2 & 4 \\ \hline 3 & 5 \\ \cline{1-2}\end{array} \quad \begin{array}{|c|c|c|}\hline 1 & 3 & 4 \\ \hline 2 & 5 \\ \cline{1-2}\end{array}$$

Figure 33.4: All standard Young tableaux of shape (3,2)

Unfortunately, no simple combinatorial proof of the hook formula exists. A number of outlines of existing proofs are given by Sagan ([389], p. 266).

Young tableaux are related to the permutation group by a construction called the Schensted correspondence, or perhaps more appropriately, the Robinson-Schensted-Knuth correspondence. For a more in-depth discussion of the nomenclature, see Fulton ([158], p38). The Schensted correspondence is a bijection between the set of permutations and ordered pairs of Young tableaux. Interested readers should consult Sagan [389] or Stanton and White [454] for more general treatments of the correspondence.

Theorem 33.3. (Schensted correspondence).

There exists a bijection between permutations $\pi \in S_n$ and all pairs of standard Young tableaux (P, Q) of the same shape $\lambda \vdash n$.

Proof. We will construct the pair (P, Q) from a permutation π to show that any permutation can be mapped onto an ordered pair of Young tableaux. We will then provide a means of recovering π from the pair (P, Q) to prove every ordered pair corresponds to a unique permutation. In the spirit of the previous section, we will say that the m-th *card* in our permutation is $\pi(m)$.

Assume π is a permutation of the numbers $1, 2, \ldots, n$. It will help to think of π in a two-line form where the first line are the integers from 1 to n and the second line is the permutation.

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 8 & 3 & 7 & 9 & 2 & 5 & 4 & 1 & 10 & 6 \end{pmatrix} \tag{33.2}$$

We will construct the Q-tableau from the first line and the P-tableau from the second by inserting one card at a time moving from left to right. Denote by P_m the tableau created after inserting m cards, and Q_m analogously. To insert the m-th card, we use the following rules.

1. If $\pi(m)$ is larger than all cards in the current column of P_{m-1}, append $\pi(m)$ to the end of the current column.

2. If $\pi(m)$ is not larger than all cards in the current column of the P_{m-1}, replace the smallest card b such that $b > \pi(m)$ with $\pi(m)$. Now use rules 1 and 2 to insert b into the next column to the right.

3. After $\pi(m)$ has been inserted into the P_{m-1}, we obtain P_m, which has the same shape as P_{m-1} except for a single added cell. Create Q_m by adding that same cell to Q_{m-1} with the number m in that cell.

As we can see, this algorithm produces two standard Young tableaux of the same shape from the permutation π. To give an example, we construct the two tableau for the permutation in (33.2) above in Figure 33.5.

Now we must describe how to recover a permutation π from a pair of standard Young tableaux. We essentially perform the reverse operation to the column insertion described in the first half of the proof. Again, we let P_m be the P-tableau with n cells, and Q_m the Q-tableau with m cells. The procedure is outlined below:

1. Remove the largest entry m in Q_m to form Q_{m-1}. Find the corresponding entry b in P_m and remove it from the tableau.

2. If b is in the first column of P, then set $\pi(m) = b$.

3. If b is not in the first column, we insert b into the column to the left. Find the largest entry c of this column such that $b > c$. Put b in the position of c and then place c according to rules 2 and 3.

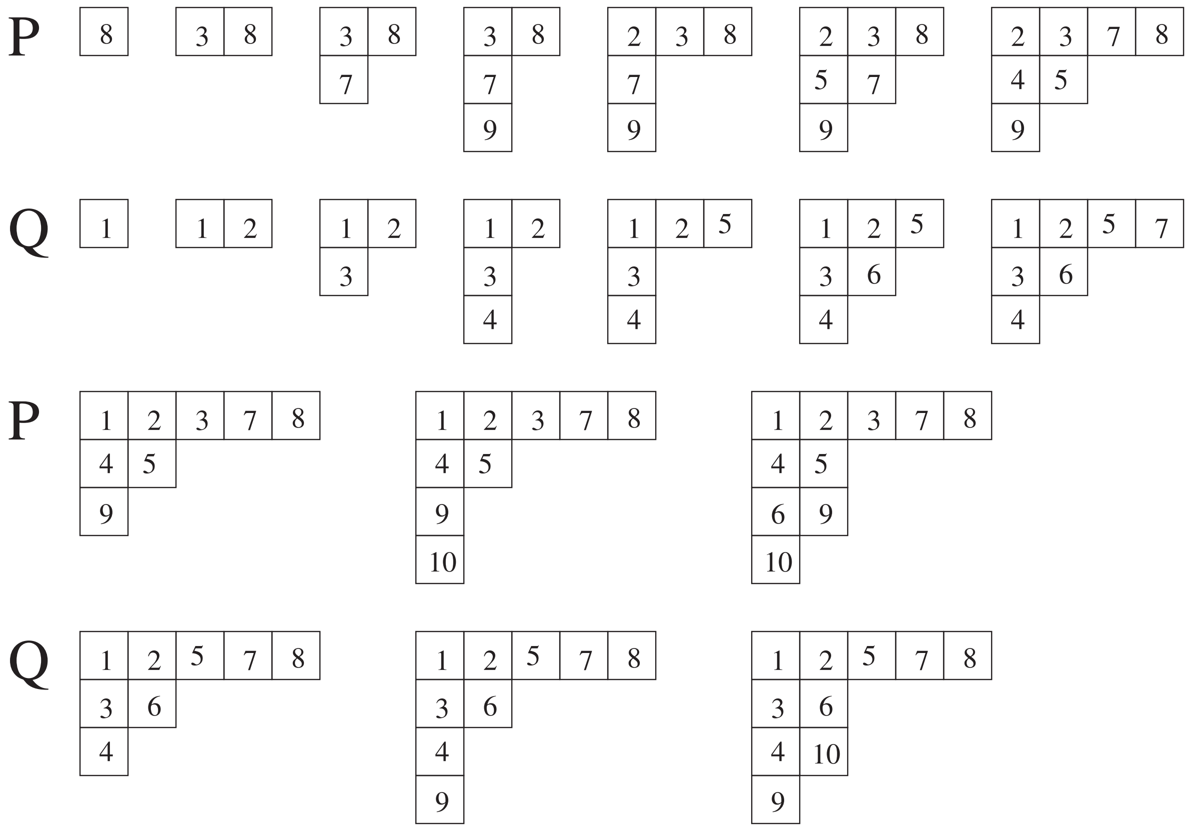


Figure 33.5: The Schensted correspondence for the example permutation in (33.2)

It is clear that this procedure simply reverses the previous construction. Figure 33.6 provides an example of recovering π from a pair of Young tableaux. The reader can verify that the given permutation recovered does generate the Young tableaux.

We have exhibited an map which takes each permutation to an ordered pair of Young tableaux. The inverse takes any pair of tableaux and sends it to a permutation in S_n. $\square$

The Schensted correspondence has a number of interesting and beautiful properties, two of which are quickly described here to give the reader the feel for how useful it is.

Proposition 33.4. *If π corresponds to (P, Q) under the Schensted correspondence, then π^{-1} corresponds to (Q, P).*

This is not too difficult to prove, and a good solution is provided in [454]. An *involution* of $\{1, 2, \ldots, n\}$ is an element π of S_n such that $\pi = \pi^{-1}$. Thus Proposition 33.4 gives us the following corollary.

Corollary 33.5. *The number of involutions of $\{1, 2, \ldots, n\}$ is $\sum_\lambda d_\lambda$.*

Schensted's original motivation in constructing this correspondence was to investigate ways of computing $l(\pi)$ for a given permutation. While this method is more tedious to perform than patience sorting, This result is more useful to us because it will allow us, via representation theory, to connect the distribution of L_n to random matrices.

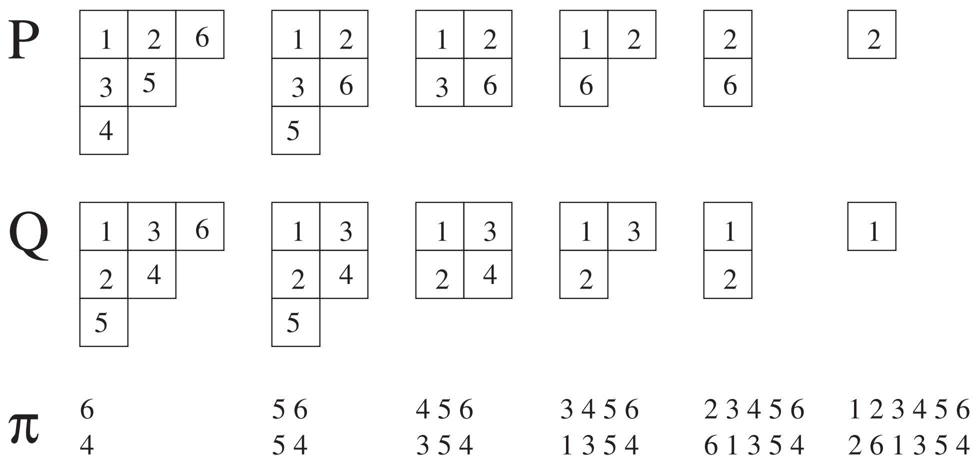


Figure 33.6: The Schensted correspondence from tableaux to permutation

Proposition 33.6. *Given a permutation $\pi \in S_n$, the number of rows in the corresponding P-tableau given by the Schensted correspondence is equal to $l(\pi)$. The length of the longest decreasing subsequence is given by the number of columns of the P-tableau.*

We can now give an explicit formula for the distribution of L_n by counting the number of Young tableaux with a certain number of rows.

$$P(L_n = l) = \frac{1}{n!} \sum_{\lambda \vdash n, |\lambda| = l} (d_\lambda)^2 \tag{33.3}$$

33.5 Representations of S_n

In this section we review some basics of representation theory and describe the link between the representations of the symmetric group S_n and standard Young tableaux. For a more basic treatment of group representations the reader is referred to the algebra textbook by Artin [20]. For more about representations of the symmetric group, the books Diaconis [111] and Sagan [390] are good overviews.

Let G be a group and V be a finite-dimensional vector space over the complex numbers $\mathbb{C}$. Then a *representation of* G is a homomorphism $\rho : G \to GL(V)$, where $GL(V)$ is the general linear group of all invertible linear transformations from V to itself. The space V is called the G-module associated with ρ because G acts on the space V. The *degree* of ρ is defined to be $\dim(V)$.

The representation $\rho : G \to GL(V)$ that sends all elements of G to the identity is called the *trivial representation* and has degree 1. If $G = S_n$, then we can let ρ map G to the set of $n \times n$ permutation matrices. This is called the *defining representation* of S_n and has degree n.

A G-module V is called *irreducible* if there is no proper subspace W of V that is invariant under the action of G. Mashke's theorem states that every G-module can be written as the direct sum of irreducible G-modules. The number of irreducible representations is equal to the number of conjugacy classes of G.

Consider the group S_n. The *cycle-type* of a permutation is a list $(\lambda_1, \lambda_2, \ldots, \lambda_k)$ of the cycle lengths in the permutation. The conjugacy classes of S_n consist of permutations having the same cycle type. Note that a cycle-type is the same as a partition of n. We can in fact identify with each partition $\lambda \vdash n$ an irreducible representation S^λ of S_n called a *Specht module*.

The *character of a representation* ρ is a function $\chi : G \to \mathbb{C}$ defined by $\chi(g) = tr(\rho(g))$. The character is constant on the conjugacy classes of G. Let $e \in G$ be the identity element. Then every representation must map e to the identity matrix in $GL(V)$. Thus $\chi(e) = tr(\rho(e)) = \dim(V)$. In the case where $G = S_n$ we denote the characters of S^λ by χ^λ. The most important fact for us we shall state without proof. Readers who are interested should consult [158] or [390].

Lemma 33.7. *Let λ be a partition of n, S^λ be the corresponding Specht module, and e be the identity element of S_n. Then:*

$$d_\lambda = \dim(S^\lambda) \tag{33.4}$$

This is a powerful result which relates the combinatorial enumeration of standard Young tableaux to the representations of the symmetric group. It follows from a construction which turns the standard Young tableaux of shape λ into a basis for the Specht module S^λ. Combining Lemma 33.7 and equation 33.3 we obtain the following expression for the distribution of L_n.

$$P(L_n \leq l) = \frac{1}{n!} \sum_{\lambda \vdash n, |\lambda| \leq l} (\chi^\lambda(e))^2 \tag{33.5}$$

33.6 Random matrices and L_n

We are now ready to connect the results from the previous sections to what we know about random matrices. We have expressed the density of L_n in terms of the characters of the symmetric group. Frobenius showed an explicit relation via power-sum symmetric functions between the characters of the symmetric group and the characters of the unitary group. By exploiting this relation, we can express the quantity in equation (33.5) in terms of an inner product of two power-sum symmetric functions, which is also a way of calculating a matrix integral over the unitary group. In this section we will let $k = |\lambda|$ to make the notation clearer.

The power-sum symmetric functions play an important role in the following analysis, since they are the link between random matrices and random permutations. Let U be a $k \times k$ matrix with eigenvalues $\mathbf{x} = (x_1, x_2, \ldots, x_k)$. Then the power sum symmetric function P_j is given by:

$$P_j(U) = \sum_{i=1}^{n} x_i^j$$

If $\lambda \vdash n$, then we have:

$$P_\lambda = \prod_{j=1}^{k} P_{\lambda j}$$

The most useful fact about these function is their relationship to the Schur functions. In fact, Frobenius showed that the Schur functions form a basis for the power-sum symmetric polynomials:

$$P_\lambda = \sum_{\mu \vdash k} \chi^\lambda(\mu) s_\mu$$

where $\chi^\lambda(\mu)$ is the character of the Specht module S^λ of the symmetric group S_n, and the Schur functions $\{s_\mu | \mu \vdash k\}$ are the characters of the unitary group $U(k)$ of $k \times k$ matrices.

Diaconis and Shahshahani [117] give an expression for the inner product of two power-sum symmetric functions, where the expectation is taken over the unitary group with normalized Haar measure.

$$E_{U\in U(l)}[P_\lambda(U)\overline{P_\mu(U)}] = \frac{1}{n!}\sum_{\nu,\kappa\vdash n}\chi^\nu(\lambda)\chi^\kappa(\mu)E_{U\in U(l)}(s_\mu(U)s_\kappa(U)) \tag{33.6}$$

We are now ready to prove our the main result of this paper.

Theorem 33.8. *The cumulative distribution function of the random variable L_n is given by the following equation.*

$$P(L_n \le l) = \frac{1}{n!}\int_{U(l)} |Tr(U)^n|^2 \mathrm{d}U \tag{33.7}$$

where $\mathrm{d}U$ is the normalized Haar measure on the group $l \times l$ unitary matrices.

Proof. The power-sum symmetric function $P_1(U)$ is simply the trace of U. If $\lambda = 1^n$, then $P_\lambda(U) = Tr(U)^n$. So we can see then that the integral in the right-hand side of the result is the inner product of $P_{1^n}(U)$ with itself. Thus we can set $\lambda = \mu = 1^n$ in (33.6) to obtain a a much simpler expression.

$$\int_{U(l)} |Tr(U)^n|^2 \mathrm{d}U = \sum_{\nu\vdash n, k\le l}(\chi^\nu(1^n))^2(s_\nu(U)s_\nu(U)$$

Because the Schur functions s_μ are orthogonal, those terms vanish, and we are left with:

$$\int_{U(l)} |Tr(U)^n|^2 \mathrm{d}U = \sum_{\nu\vdash n, k\le l}(\chi^\nu(1^n))^2$$

But this is the same expression as in (33.5), since the partition 1^n corresponds to the identity permutation e. The result follows immediately. □

33.7 Conclusion

There are a number of questions we can ask at this point. What is the asymptotic behavior of L_n as $n \to \infty$? Are there similar interpretations for matrix integrals over the orthogonal and symplectic group? Can we obtain similar results for colored permutations and multiset permutations?

As it turns out, the asymptotic behavior of L_n has been studied by many mathematicians, and a good summary of the research from many different perspectives can be found in the paper by Aldous and Diaconis [2]. For extensions to other classical groups, the paper by Rains [371] relates results on those groups to problems in counting longest increasing subsequences with some restrictions, as well as some results on colored permutations. Odlyzko and Rains [338] have explicit results and more in-depth Monte Carlo simulation for the behavior of L_n.

33.8 Computing $l(\pi)$ with patience sorting

Below is code in MATLAB used to compute a histogram of $l(\pi)$.

```
%% patienceSort.m
%%
%% Performs greedy patience sorting on decks of length n
%% for given number of samples.  The output histogram is
%% saved in v.

n = 10;
samples = 1e7;

v = zeros(n,1);

for loop1 = 1:samples
   ord = randperm(n);
   piles = n;

   for ind1=1:length(ord)
     curr = ord(ind1);

     [val, pos] = find(piles > curr);

     if (val ~= [])
       piles(min(pos)) = curr;
     else
       piles = [piles curr];
     end
   end

   v(length(piles)) = v(length(piles)) + 1;
end
```

Code 33.1

33.9 Constructing the Schensted correspondence

33.10 Computing $l(\pi)$ with matrix integrals