A walk through Combinatorics

Toan Q. Pham
mathtangents@gmail.com

Friday 16th November, 2018

Todo list

1 Chapter 5: Divide and Conquer. Partitions

Some terminology. Partition of $[n]$ into *blocks*. Partition of n into *parts*. Notation for partition of n is $\pi = (\pi_1, \ldots, \pi_t)$.

Ferrers diagram: *conjugate*, *hook*.

Useful table 5.1 from [1].

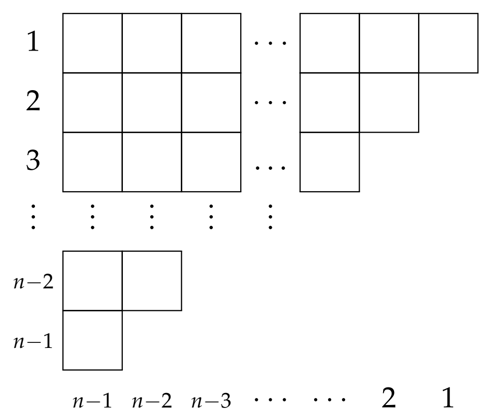
Definition 1.0.1 (Set partitions)**.** A partition of set $[n]$ is a collection of non-empty blocks so that each element of n belongs to exactly one of these blocks. The number of partition of n into k nonempty blocks is denoted by $S(n,k)$. The number $S(n,k)$ is called the *Stirling number of the second kind*.

Definition 1.0.2 (Integer partitions)**.** Let $a_1 \geq a_2 \geq \cdots a_k \geq 1$ be integers so that $a_1 + \cdots a_k = n$. Then the sequence $(a_1, \ldots, a_k)$ is called a *partition* of the integer n. The number of all partitions of n is denoted by $p(n)$. The number of partitions of n into exactly k parts is denoted by $p_k(n)$.

Problem 1.0.3. (a) Let $h(n)$ be the number of ways to place any number (including zero) non-attacking rooks on the Ferrers shape of the "staircase" partition $(n-1, n-2, \ldots, 1)$. Prove that $h(n) = B(n)$.

(b) In how many ways can we place k non-attacking rooks on this Ferrers shape.

Proof. (a) We label the rows and columns of the Ferrers diagram as follow:



In other words, each box has a coordinate (i,j) where $1 \leq i \leq j \leq n-1$. We will create a bijection f from "$h(n)$" to "$B(n)$" (they technically are not sets but for the sake of convenience).

Consider an arbitrary partition of $[n]$, consider the parts that don't contain n. In each such part, we can order the elements in increasing order: $a_1 < a_2 < \cdots < a_k$. Now for each i in $[k-1]$, we put a rook in (a_i, a_{i+1}). For parts that have only one element $i \neq n$, we put a rook in (i,i).

We need to show that this can map to "$h(n)$". Indeed, from the mapping f, we can see that there exists at most one rook that has the coordinate (i,j). Thus, f does indeed map to "$h(n)$".

Next, we show that f is bijective. This is quite obvious since for any different partitions of $[n]$, there exists one coordinate that cannot belong to both partitions, which results in different ways of putting the rooks.

We show that f is surjective. Indeed, if there is a rook in (i,j) then we put two elements i,j in same part that does not contain n. If there is no rook in row i then that implies i and n are in the same part.

Thus, $h(n) = B(n)$.

(b) It seems the bijection in (a) does not yield nice result as $S(n, n-k)$. See the bijection from the book for this answer. (It's essentially the same as (a) except the label for the columns is $n, n-1, \dots, 2$ instead of $n-1, n-2, \dots, 1$). The bijection of the book is able to do this since for each rook drawn, two blocks are combined. This is not true for the bijection in part (a). □

Problem 1.0.4. Let m, n be positive integers such that $m \geq n$. Prove that the Stirling number of the second kind satisfy the recurrence relation

$$S(m,n) = \sum_{i=1}^{m} S(m-i, n-1) n^{i-1}.$$

$S(m-i, n-1)n^{i-1}$ is number of ways to partition $[m]$ into m parts where the block π that contains i does not contain other numbers less than i and the remaining $n-1$ blocks each must have one number less than i. Summing all up, the statement follows.

Problem 1.0.5. Prove that the number of partitions of n into exactly k parts is equal to the number of partitions of n in which the largest part is exactly k.

This is just Ferrers diagram taking conjugate.

Problem 1.0.6. Prove that the number of partitions of n into at most k parts is equal to that of partitions of $n+k$ into exactly k parts.

Consider a Ferrer diagram/shape of the partition of n into at most k parts, we add a column of k boxes into the left side of this diagram then we've created a Ferrers diagram of partition of $n+k$ into exactly k parts.

Problem 1.0.7. The Durfee square of a partition p is the largest square that fits in the top left corner of the Ferrers shape of p. If we know the parts of a partition p, how can we figure out the side length of its Durfee square without drawing the Ferrers shape of p?

For a partition $p = (p_1, \dots, p_t)$. The length of the Durfee square is the largest i such that $p_i \geq i$.

Problem 1.0.8. Let k be a positive integer, and let q be a non-negative integer such that $q < k$. Define $p_{k,q}(r) = p_k(rk+q)$. Prove that $p_{k,q}(r)$ is a polynomial function of r.

Proof. Consider a partition $p = (p_1, \dots, p_k)$ of n into k parts. Since $q < k$ so $1 \leq p_k \leq r$ If $p_k = i$ then there are $p_{k-i,q}(r-i)$ such partitions □

2 Chapter 6: Cycles in Permutations.

Definition 2.0.1 (Stirling number of the first kind)**.** The number of n-permutations with k cycles is called a signless Stirling number of the first kind, and is denoted by $c(n,k)$. The number $s(n,k) = (-1)^{n-1}c(n,k)$ is called a *Stirling number of the first kind*.

Definition 2.0.2. *k-cycle* means cycle of length k.

If an n-permutation p has a_i cycle of length i, for $i = 1, 2, \ldots, n$ then we say $(a_1, \ldots, a_n)$ is the *cycle type* of p.

For a permutation $p = p_1 p_2 \cdots p_n$ of $[n]$, then $q = q_1 \cdots q_n$ where $q_i = n + 1 - p_i$ is called the *complement* of p.

Canonical form: Entries of permutation q of $[n]$ that are larger than all entries on their left are called *left-to-right maxima*. This corresponds to the leading entry in the canonical form of permutations.

Interesting relation between Stirling numbers of fist kind and of second kind.

perm_intepretation **Remark 2.0.3** (Ways to view permutation)**.** Different ways to view permutation have interesting application. Two types of notation learned from the book is: canonical cycle notation and one-line notation.

One advantage of using canonical form for permutations is that we know the positions of the numbers and that help for reference, e.g. create procedure that involves certain positions. However, it relies on the comparison between numbers which we usually don't need when talking about permutation. Hence, there is a second interpretation of permutations:

Each cycle from p can be seen as a circle with numbers on the circle in the same order as the cycle. For example, if 1 is on a circle then $p(1)$ will be the number on the right of 1 on that circle. Note that for counting permutation, the order of the numbers in each circle matters but the order of the circles does not matter.

Remark 2.0.4. Symmetry: Permutation has a bunch of "symmetry". Hence, if you want to prove some local property (i.e. in certain structure or sets) about permutations then such property may also true globally (i.e. for any such structures or sets). Indeed, this can be seen from following example

prop:bona_chap6_18 **Proposition 2.0.5.** Let i and j be two elements of $[n]$. Then i and j are in the same cycle in exactly half of all permutations of $[n]$.

Slick proof. WLOG, suppose $i < j$. For each n-permutation such that i and j are in the same cycle, we can define a bijective mapping f by swapping i with $n-1$ and j with n to get a n-permutation with $n-1$ and n being in the same cycle.

On the other hand, consider the canonical form of permutation then n and $n-1$ in the same cycle iff $n-1$ is on the right of n, which happens for exactly half number of permutations. $\square$

Counting proof. WLOG, suppose $j < i$. We will use the canonical cycle notation. Note that the first element of the cycle that contains j must be at least j.

If that first element is some $k > i$. In order for k to be the first element of a cycle, all numbers at the left of k must be less than k. In order for i, j to be in the same cycle whose first element is

k, the numbers i, j must be in between k and a number that is larger than k and is closest to k. In order to satisfy these conditions, here are our procedure:

1. Arrange $n-k+1$ numbers $k, \ldots, n$ such that k is first position. There are $(n-k)!$ ways to do that.

2. Put i, j right after k. There are $2! = 2$ ways do that.

3. We left with $k-3$ numbers that are less than k. There are $(n-k+4)\cdots n$ ways to put these numbers in.

Hence, there are $(n-k)! \cdot 2 \cdot (n-k+4) \cdots n = \frac{2n!}{(n-k+1)(n-k+2)(n-k+3)}$ permutations such that i, j are in the same cycle whose first element is some $k > i$.

If $k = i$ then similarly, we obtain $(n-i)! \cdot 1 \cdot (n-i+3) \cdots n = \frac{n!}{(n-i+1)(n-i+2)}$.

Summing up, we obtain

$$\frac{n!}{(n-i+1)(n-i+2)} + \sum_{k=i+1}^{n} \frac{2n!}{(n-k+1)(n-k+2)(n-k+3)} = \frac{1}{2}n!.$$

□

:perm_gap_position **Definition 2.0.6** (Gap positions)**.** For a permutation p of $[n]$ in canonical form, p has $n+1$ *gap positions*, one *after* each element in each cycle, and one at the very beginning of the permutation, before all entries and cycles. For example, permutation $(42)(513)$ has six gaps positions, indicated by the bars in the following array: $|(4|2|)(5|1|3|)$.

One can see gap positions in a different view: Each cycle of the permutation p can be represented as a circle with the numbers of the cycle written in order on that circle. For each such circle of length i we see i gaps between any consecutive numbers in the cirle. Furthermore, there is an extra gap that does not belong to any circle. Summing up, we get $n+1$ gaps for a permutation of length n.

The first interpretation allows us to label the gap positions as $1, 2$ to $n+1$. The second interpretation emphasises the "balance" between the gap positions, i.e. each gap position is essentially between two (not necessarily the same) numbers. This helps us to construct new permutations using gap positions. For example, if we replace a gap position between a_1 and a_2 with b then we can create new permutation p with $p(a_1) = b, p(b) = a_2$.

Lemma 2.0.7 (Transition lemma)**.** Let $p : [n] \to [n]$ be the permutation written in canonical cycle notation. Let $g(p)$ be the permutation obtained from p by omitting the parentheses by reading the entries as a permutation in the one-line notation. Then g is the bijection from the set S_n of all permutations of $[n]$ onto S_n.

bona_walk_to_com

lem:bona_chap6_20 **Problem 2.0.8** (From [1], lemma 6.20)**.** Let ODD(m), resp. EVEN(m) be the set of m-permutations with all cycle lengths odd, resp. even.

For all positive integer m, the equality $|ODD(2m)| = |EVEN(2m)|$ holds.

Since it's $2m$ so any $2m$-permutation in $ODD(2m)$ has even number of cycles. With this information, we know more about $ODD(2m)$ than $EVEN(2m)$. As a result, we decide to find a bijection **from** $ODD(2m)$ to $EVEN(2m)$ in the hope that this fact can be useful.

Let $C_1, \dots, C_{2k}$ be the cycles of odd length in canonical form. We want to make a change to these cycles so that each has even length. Observe that a local change (i.e. to certain cycles) cannot cause a global change to all the cycles. For example, vaguely, if you pick an element in C_i and put it in a position in some C_j then the cycles C_k's $(k > \max\{i, j\})$ remain unchange.

Hence, we need a global change (i.e. for each cycle).

Attempt 2.0.9. One thought comes to mind is to take out one element from each cycle and combine these elements to create a new cycle. This of course will create a new permutation with all cycle lengths even. However, does this gives us an injection to $EVEN(2m)$? To answer that, we need to specify our bijection.

How would you choose an element from each cycle C_i? Let's try to guess some simple way to choose the element: If we choose largest element in each C_i, i.e. the first element in each C_i (canonical form) then we would break the arrangement of the cycle and hence, we don't know if the remaining elements form a cycle of even length in canonical form or not. What if we choose the smallest element in each C_i? Since we don't know if this is valid or not, let's turn to the second question:

How would you order the chosen elements to create a new cycle in canonical form? There are many ways to arrange the elements to create a new cycle. On the other hand, we can't seem to extract any information that can help us to pick a particular permutation. After taking some trials and errors for small m, we notice that it's quite hard to give an injective map. For any procedure you choose to arrange the elements, the resulting map is likely not injective.

Thus, this idea will likely not work.

Observation 2.0.10. One of our main idea to create cycles of even length is to move elements from one cycle C_i to the other. We don't want to move more than 2 elements in each cycle, because this implies that there are cycles of length at least 2, which is not always possible. Hence, we can move at most one element from each cycle.

Observation 2.0.11. If we move one element from one cycle to the other then there must be half of the cycles that receive element and the other half giving out elements. This agrees with our observation that there are even number of cycles in each permutation in $ODD(2m)$. This makes us more confident about this method.

What we left with is to find a systematic way to move the elements. There are some ways to do that and you just have to choose the simplest way, i.e. move an element from C_{2i-1} to C_{2i}. Even more specific, we take the last element from C_{2i-1} and put it in the end of C_{2i}. Checking with some small examples of m, we are more confident that this is a bijection.

In the end, we obtain the following proof.

Proof. See [1], page 121. We construct a bijection $\Phi : \text{ODD}(2m) \to \text{EVEN}(2m)$. □

theo:bona_chap6_24 **Problem 2.0.12.** Let $\text{ODD}(m)$, resp. $\text{EVEN}(m)$ be the set of permutation for $[m]$ with all cycle lengths odd, resp. even. For all positive integer m then

$$|\text{ODD}(2m)| = |\text{EVEN}(2m)| = 1^2 \cdot 3^2 \cdots (2m-1)^2.$$

Thoughts and proof. Since the right-hand side looks quite nice so let's try to find an elegent procedure for this.

First, observe that the right-hand side is the product of $2m$ numbers. What does $2m$ refers to? It's is the number of elements in the set $[2m]$ or the number of positions that we can put the elements in. Hence, we can predict that our procedure will involve either choosing one element at a time or choosing one position at a time.

Before going into the details, let's list all the permutation in $\text{EVEN}(2m)$ for $m = 2$ first:

$$\begin{array}{ccc} (4321) & (4312) & (4213) \\ (4231) & (4123) & (4132) \\ (32)(41) & (21)(43) & (31)(42) \end{array}$$

We want $|\text{EVEN}(4)| = 3^2$. We want to use the multiplication rule so the answer 3^2 motivates us to partition $\text{EVEN}(4)$ into three sets of equal size. How would you partition the sets?

Let's go back to our first observation. If our procedure is to choose one element at a time, let's check if this gives you the desired partition $\text{EVEN}(4)$. If we choose 1 first, then there are three possible positions for 1 (1 cannot be in the first position), which parition $\text{EVEN}(4)$ into three unequal sets

$$\{4123, 4231, 2143, 3142\}, \{4321, 4231\}, \{4312, 4213\}.$$

Hence, we don't want "choosing 1" as our first step in the procedure. If you check with "choosing 2" or "choosing 3" or "choosing 4", you will find that it does not give what we want. Thus, we will temporary abandon the idea of choosing the elements for our procedure.

Let's go to the next possible idea for the procedure, which is to choose one position at a time. If you our first step is to choose element for the first position then there are 3 possible elements for the first positions (1 cannot be in the first position) which parition $\text{EVEN}(4)$ into three sets of unequal size

$$\{4321, 4312, 4213, 4231, 4123, 4132\}, \{3241, 3142\}, \{2143\}.$$

Similarly, you can check to set what happens when when we find element for the ith position first. Finally, we notice that "choosing element for the 4-th position" gives the desired partition. Element in the 4-th position cannot be 4 so we can partition $\text{EVEN}(4)$ into three sets

$$\{4321, 3241, 4231\}, \{4213, 4123, 2143\}, \{4312, 4132, 3142\}.$$

Within each set, with a completely similar argument as above, we find the next step for the procedure: "choose element for the 3-th position" (observe that after putting an element in the last position, there are 3 choices for the 3-th position).

From the example, we find that this idea can be generalized which gives us the factor $(2m-1)^2$ for $\text{EVEN}(2m)$. On the other hand, we know that $1^2 \cdot 3^2 \cdots (2m-3)^2 = |\text{EVEN}(2m-2)|$. This suggests that the idea of after choosing elements for $2m-1$th and $2m$th position, the remaining $2m-2$ positions can be an element from $\text{EVEN}(2m-2)$. This is indeed true. This solves the problem. □

Proof in a different view. We can see the cycles of a permutation p in $\text{EVEN}(2m)$ as circles with numbers labelled on each circle as mentioned in remark 2.0.3. With 1 in some circle, there are $2m-1$ choices (except 1 itself) for $p(1)$, which is an element next to 1. After choosing $1, p(1)$

(with $1 \neq p(1)$), since $p \in \text{EVEN}(2m)$ so $p^2(1)$ can be equal to 1 so there are $2m-1$ choices (except $p(1)$) for $p^2(1)$. With this, we've identified the factor $(2m-1)^2$. We can proceed similarly from here. □

theo:bona_chap6_25 **Problem 2.0.13** ([1], theorem 6.25)**.** Show that for all positive integers m then

$$|\text{ODD}(2m+1)| = (2m+1)\cdot|\text{ODD}(2m)| = 1^2\cdot 3^2\cdots(2m-1)^2(2m+1).$$

The thinking for this problem is completely similar to the previous example. By listing all elements in ODD(5), we observe that the event "choose element for the last position" gives a good partition to ODD(5). Here is the final proof:

Proof. We prove by induction on m. Consider the canonical form of permutations. For $\text{ODD}(2m+1)$, there are $2m+1$ ways to an element for the last position. If the last position is $2m+1$ then the first $2m$ positions will be a permutation in $\text{ODD}(2m)$, which is what we want.

If the last position is not $2m+1$, then the $2m$-th position cannot be $2m$, which follows there are $2m-1$ ways to choose an element for the $2m$-th position. After doing this, the first $2m-1$ positions will form a permutation in $\text{ODD}(2m-1)$. This gives $(2m-1)\text{ODD}(2m-1) = \text{ODD}(2m)$ ways to choose the first $2m$ positions given that the last position is not $2m+1$.

Thus, if we add all the cases, we obtain $|\text{ODD}(2m+1)| = (2m+1)\cdot|\text{ODD}(2m)|$. □

Remark 2.0.14. In the previous three problems, when using canonical form of permutations, we noticed that our procedure always involve "choosing one position at a time" rather than "choosing one element at a time". The main reason for our option is that the event of "choosing one element at a time" does not give a nice partition for our set so that we can see the symmetry. This can be vaguely explained from the canonical form of permutations. In particular, the canonical form of permutations orders the elements by comparing them to each other. This follows that if we partition our set depending on some particular element, the symmetry may not hold.

Alternative bijective proof. This is given in [1, Theorem 6.25].

Our main idea is to use bijection $\Phi : \text{ODD}(2m) \to \text{EVEN}(2m)$ defined in problem 2.0.8 to construct a bijection Ψ from $\text{ODD}(2m)\times[2m+1] \to \text{ODD}(2m+1)$.

Here are the steps:

1. Pick $\pi \in \text{ODD}(2m)$ and $k \in [2m+1]$.
2. Apply Φ to π to get $\Phi(\pi) \in \text{EVEN}(2m)$.
3. Put $2m+1$ in the k-th gap position of $\Phi(\pi)$ (use the first intepretation 2.0.6 for gap positions to label the positions, but then use the second intepretation to put $2m+1$ in). The cycle C containing $2m+1$ will then have odd length.
4. Keep C the same but apply Φ^{-1} to the remaining even cycles of $\Phi(\pi)$ to obtain a new permutation p of all odd cycles (including C). Hence, $p \in \text{ODD}(2m+1)$.

Thus, for each $(\pi, k) \in \text{ODD}(2m)\times[2m+1]$, we are able to construct $p \in \text{ODD}(2m+1)$.

It's obvious Ψ is injective. It suffices to show Ψ is surjective. This can be achieved by identifying Ψ^{-1}, which is essentially the reverse of above steps. In particular, we have

1. Given $\pi \in \text{ODD}(2m+1)$ and identify cycle C in π that has entry $2m+1$.
2. The remaining cycles form a element in $\text{ODD}(2\ell)$ for some ℓ. Run the remaining cycles through Φ to obtain an new permutation π' with all even cycles.
3. Remove $2m+1$ from C (but remember the gap position). Combining with π', we get a new permutation from $\text{EVEN}(2m)$. Record the gap position k (where $2m+1$ was previously put) with respect to π'.
4. Run π' through Φ^{-1} to get $\Psi^{-1}(\pi)$.
5. Obtain $\Psi^{-1}(\pi)$ and gap position $k \in [2m+1]$.

$\square$

Remark 2.0.15. Observe the importance of applying Φ to switch back and forth ODD and EVEN.

2.1 Exercises

lem:polynomial_degree **Lemma 2.1.1.** Let $p(k)$ be a polynomial of degree d. Prove that $q(n) = \sum_{k=1}^{n} p(k)$ is a polynomial of degree $d+1$. Prove that this polynomial satisfies $q(0) = 0$.

This follows that a function $f(n)$ is a polynomial of degree d iff the function $g(n) = f(n) - f(n-1)$ is a polynomial of degree $d-1$.

Problem 2.1.2. Prove that for any fixed k, the function $c(n, n-k)$ is a polynomial function of n. What is the degree of that polynomial.

Proof. Induction on k. It suffices to show that $c(n, n-k) - c(n-1, n-1-k)$ is a polynomial of n. We have $c(n, n-k) - c(n-1, n-1-k) = (n-1)c(n-1, k-1)$. The right-hand side is lem:polynomial_degree a polynomial of n so the left-hand side is also a polynomial of n. From lemma 2.1.1, we find $c(n, n-k)$ is a polynomial of n. To find the degree of $c(n, n-k)$, say the degree is $f(k)$. From the above identity, we find $f(k) - 1 = 1 + f(k-1)$ so $f(k) = 2k$. $\square$

Problem 2.1.3. Let $r(n)$ be the number of n-permutations whose square is the identity permutation. Prove that if $n \geq 1$ then $r(n+1) = r(n) + nr(n-1)$ where $r(0) = 1$.

Proof. Observe that $r(n)$ is the number of n-permutations with cycles of lengths 1 or 2. This follows in canonical form for permutations, in $r(n+1)$, the element $n+1$ must be in the last two positions.

If $n+1$ is in the last position then there are $r(n)$ ways to arrange n elements in the first n positions.

If $n+1$ is in the n-th positions then there are n ways to choose an element for the last position and $r(n-1)$ ways to arrange elements in the first $n-1$ positions. $\square$

Problem 2.1.4. Find a recursive formula for the number $t(n)$ of n-permutations whose cube is the identity permutation.

Completely similar to previous problem. For such permutations, all cycles must be 1-cycles or 3-cycles. If the entry $n+1$ forms a 1-cycle then the remaining n entries can form a good permutation in $t(n)$ ways. If the entry $n+1$ is part of a 3-cycle, then there are $n(n-1)$ choices for other two entries of that 3-cycle, then there are $r(n-2)$ ways for the remaining $n-2$ entries to form a good permutation. We obtain $t(n+1) = t(n) + n(n-1)t(n-2)$.

Problem 2.1.5. Prove that on average, permutations of length n have H_n cycles, where $H_n = \sum_{i=1}^{n} \frac{1}{i}$.

Proof. For a given k-cycle, there are $(n-k)!$ permutations that contains this cycle. There are $\binom{n}{k}(k-1)!$ possible k-cycles, so the average number of cycles for a permutation of length n is

$$\frac{1}{n!}\sum_{k=1}^{n}\binom{n}{k}(k-1)!(n-k)! = H_n.$$

□

Problem 2.1.6. How many n-permutations contain entries $1,2,3$ in the same cycle?

Proof. Let $T(1,2,3)$ be the set of permutations. If f be a mapping from $T(1,2,3)$ to $T(n-2,n-1,n)$ by swapping 1 with $n-2$, 2 with $n-1$ and 3 with n. Note that f is a bijection.

It suffices to find $T(n-2,n-1,n)$. We back to our canonical form. This happens when $n-2, n-1$ are positioned on the right of n. There are 3! ways to permute $\{n-2,n-1,n\}$ and we want two permutations $(n,n-1,n-2)$ and $(n,n-2,n-1)$. This gives $\frac{n!}{3!}\cdot 2 = n!/3$. □

Problem 2.1.7. An alpine ski team has n members. They descend a particular slope one by one every day, and no two of them ever record identical times. On an average day, how many times will the best record of that day be broken?

Proof. Record the times for n members as a permutation of $[n]$, with n indicating the best record after that day. If n members ski one by one, number of times the best record of that day be broken is the number of left-to-right maximas for the permutation on that day. By canonical form of permutations, this is the number of cycles for that permutation. From previous exercise, a permutation of length n has on average H_n cycles. □

Problem 2.1.8. An airplane has n seats, and all of them have been sold for a particular flight, with no overbooking. When the last passenger arrives, he finds that his seat is taken. When he shows his reservation to the passenger at his seat, that passenger stands up, and goes to her own assigned seat. If that seat is empty, she seats down, and the seating procedure is over. If not, she shows her reservation to the person seating at that seat. That person stands up, and goes to his assigned seat, and so on. This procedure continues until someone finds his or her assigned seat empty. Tom was not the last passenger to board the plane. What is the probability that he has to move during this procedure?

Proof. The assigned seat for the i-th passenger is i but that passenger but actually seats at the $p(i)$-th seat where p is a permutation of $[n]$.

If π is the cycle of p containing n then the procedure involves this cycle π. In particular, the n-th passenger will move to seat $p(n)$, the passenser that seats on seat $p(n)$ will move to his/her assigned seat $p^2(n)$, ... Therefore, Tom has to move when Tom is in the same cycle as n. If Tom is the i-th passenger, in canonical form of permutations, this happens when i is on the left of n, which happens with probability $\frac{1}{2}$. □

Problem 2.1.9. Let p be a n-permutation. We associate a *permutation matrix* A_p to p such that $A_p(i,j) = 1$ if $p(i) = j$ and $A_p(i,j) = 0$ otherwise. Then $|\det(A_p)| = 1$.

Induction on n.

Problem 2.1.10. Prove that if p and q are two n-permutations of $[n]$, then $A_pA_q = A_{q\circ p}$.

Show $(A_pA_q)(i,j) = A_{q\circ p}(i,j)$.

Problem 2.1.11. The *inverse* of an n-permutation p is the permutation q for which $p \circ q = q \circ p = 12\cdots n$. We then write $q = p^{-1}$. Prove each permutation has a unique inverse.

The condition $q \circ p = 12\cdots n$ uniquely defines q. From this, we show q is the inverse.

Problem 2.1.12. Prove that f and f^{-1} are of the same type.

If $p_1 \cdots p_k$ is a k-cycle of f then $p_kp_{k-1}\cdots p_1$ is a k-cycle of f^{-1}. This follows f and f^{-1} are of the same cycle type.

Problem 2.1.13. What is the combinatorical meaning of A_p^T?

It is the permutation matrix of p^{-1}.

Problem 2.1.14. Assume that we know the cycle type $(a_1, \ldots, a_n)$ of an n-permutation. Determine the smallest positive integer d such that $p^d = 12\cdots n$.

Proof. For an entry i in a k-cycle of p, we have $p^d(i) = i$ when $k \mid d$. Since this is true for any k, we find the smallest d is the least common divisor of i's so $a_i > 0$. □

Problem 2.1.15. For a prime number t. Let $T(n)$ be number of n-permutations such that $p^t = 12\cdots n$ for some prime t. Then $T(n) \equiv 0 \pmod p$ for all $n \geq t$.

Proof. We have $T(n) = 1$ for all $1 \leq n < t$.

Next, we will calculate $T(t)$. If entry t forms a 1-cycle then all remaining entries also form a 1-cycle, which will give the identity permutation. If t forms a t-cycle then we have $(t-1)!$ choices for other entries of that t-cycle. This gives $T(t) = (t-1)! + 1$, which is divisible by t according to Wilson's theorem.

Next, we will find a recursive formula for $T(n+1)$ with $n \geq t$. From the previous problem, for such permutations, all cycles must be 1-cycle or t-cycles. If entry $n+1$ is part of a 1-cycles then the remaining n entries can form a good permutation in $t(n)$ ways. If the entry $n+1$ is part of a t-cycle, then there are $n(n-1)\cdots(n-t+2)$ choices for other $t-1$ entries of that t-cycle, then there are $r(n+1-t)$ ways to the remaining $n+1-t$ entries to form a good permutation. Thus, we obtain

$$T(n+1) = T(n) + n(n-1)\cdots(n-t+2)T(n+1-t),\ n \geq t.$$

Using this recursive formula, we find that $0 \equiv T(t) \equiv T(t+1) \equiv \cdots \equiv T(2t-2) \pmod t$. Therefore, by induction on $n \geq t$, we find $T(n) \equiv 0 \pmod t$ for all $n \geq t$. □

Problem 2.1.16. Let $n \geq 2$. Prove that $\det A_p = 1$ for exactly one half of all n-permutations p.

Proof. Let $q = (21)(3)\cdots(n)$ then we have $A_p A_q = A_{q\circ p}$ so $\det A_p \det A_q = \det A_{q\circ p}$. Since $\det A_q = -1$ so A_p and $A_{q\circ p}$ have determinants of opposite signs. On the other hand, since $q\circ(q\circ p) = p$ so the pairs of permutation $(p, q\circ p)$ is uniquely defined. Exactly one permutation from this pair has determinant 1. □

exer:bona_chap6_21 **Problem 2.1.17.** We say that a permutation $p \in S_n$ has a *square root* if there is a permutation $q \in S_n$ so that $q^2 = p$. Find a sufficient and necessary condition of p having a square root, in terms of its cycle lengths.

Proof. For $k \geq 0$, if $(a_1 \cdots a_{2k+1})$ is a $(2k+1)$-cycle of q then $(a_1 a_3 \cdots a_{2k+1} a_2 \cdots a_{2k})$ is a $(2k+1)$-cycle of q^2. For $k \geq 1$, if $(a_1, \ldots, a_{2k})$ is a $(2k)$-cycle of q then $(a_1, a_3, \cdots a_{2k-1})$ and $(a_2, a_4, \cdots, a_{2k})$ are two k-cycles of q^2.

From these two observations, we find that for each $1 \leq i \leq n$, $p = q^2$ must have even number of cycles with even length $2i$. This is a necessary condition for p to have a square root.

We show that the above condition is also a sufficient condition for p to have a square root q. To do this, we reverse the action from q to q^2. Indeed, for any $(2k+1)$-cycle $(b_1, \cdots, b_{k+1}, a_1, \cdots, a_k)$ in p then let $(b_1, a_1, b_2, a_2, \ldots, b_k, a_k, b_{k+1})$ be a cycle in q. For any pair of $(2k)$-cycles $(a_1, \ldots, a_{2k})$ and $(b_1, \ldots, b_{2k})$, let $(a_1, b_1, a_2, b_2, \ldots, a_{2k}, b_{2k})$ be a cycle in q. With this, we can obtain $q^2 = p$. □

exer:bona_chap6_23 **Problem 2.1.18.** Construct a bijection $\tau : \mathrm{ODD}(2m+1) \times [2m+1] \to \mathrm{ODD}(2m+2)$.

theo:bona_chap6_25 lem:bona_chap6_20

Proof. The proof is similar to Bona's proof for theorem 2.0.13. From lemma 2.0.8, we know a bijection Φ from $\mathrm{ODD}(2m)$ to $\mathrm{EVEN}(2m)$.

For any $\pi \in \mathrm{ODD}(2m+1)$, add 1 to each element to get a new permutation of $\{2, \ldots, 2m+2\}$. Next, we put 1 in kth gap position where $2 \leq k \leq 2m+2$ which will change one cycle π' from odd to even. The remaining cycles will form a permutation in $\mathrm{ODD}(2\ell)$ where 2ℓ is the length of such cycles. Run the remaining cycles through Φ to get even cycles. Combine this will π', we will get a permutation in $\mathrm{EVEN}(2m+2)$, which when applying Φ^{-1}, you will get $\mathrm{ODD}(2m+2)$. □

exer:bona_chap6_24 **Problem 2.1.19.** Let $SQ(n)$ be the set of n-permutations having at least one square root. Prove that for all positive integers n, we have $|\mathrm{SQ}(2n)| \cdot (2n+1) = |\mathrm{SQ}(2n+1)|$. Note that this means $p(2n) = p(2n+1)$, where $p(m)$ denotes the probability that a random chosen m-permutation has a square root.

From exercise 2.1.17, we know that a permutation p has a square root if it has even number of cycles with each even length $2i$. We try to impose this condition on establishing the relation between $\mathrm{SQ}(2n+1)$ and $\mathrm{SQ}(n)$.

Attempt 2.1.20. We look at $\mathrm{SQ}(5)$. Note that the event "element at the last position" gives a nice partition of $\mathrm{SQ}(5)$ into 5 disjoint subsets of equal size. However, it's difficult to figure out what to do after that, since the remaining elements do not form permutations in $\mathrm{SQ}(4)$ (which is what we want). After some failed attempts for this approach, I decided to find a new approach.

After few other attempts in applying procedures "choose elements for each position" and "choose positions for each element" for $SQ(n)$, I got nowhere. The reason for the failure lies in the property of $SQ(n)$, which is hard to use comparing to the sets $\mathrm{ODD}(n)$ or $\mathrm{EVEN}(n)$. Hence, an idea is to first apply some mapping Φ on $\mathrm{SQ}(n)$ to remove properties of $\mathrm{SQ}(n)$, then we can apply our familiar procedures.

Here is one way we can do that:

Attempt 2.1.21. What can our Φ be here? Well, since $\mathrm{SQ}(n)$ is the set of permutations that has square root, Φ may be the inverse of that procedure, i.e. map p to a square root of p. However, note that a permutation p may have multiple square roots (look at the proof of exercise 2.1.17 to see why). For example, $(1)(2)(3)(4)$ has four square roots $(1)(2)(3)(4)$, $(21)(43)$, $(31)(42)$ and $(41)(32)$. For Φ to be well defined, we need to specify our procedure of changing permutation p to its square root. From here, we can apply familiar procedures to the square roots of $\mathrm{SQ}(2n+1)$. In particular, the steps from here are similar to proof of problem 2.1.18.

First, we will go and define Φ which maps $p \in \mathrm{SQ}(2n+1)$ to square root of p. For $p \in \mathrm{SQ}(2n+1)$. If $(a_1a_3\cdots a_{2k+1}a_2\cdots a_{2k})$ is odd $(2k+1)$-cycle of p then assign $(a_1,\cdots,a_{2k+1})$ to be $(2k+1)$-cycle of square root of p. Let $C_1,\dots,C_{2k}$ be $(2i)$-cycle of p in that order when writing p in canonical form. Any pair C_{2j-1}, C_{2j} of $(2i)$-cycles will form a $(4i)$-cycles for q (as demonstrated in exercise 2.1.17). However, what does Φ actually map to? ... After a few trials, I found that this attempt does not work for me. Maybe a different Φ could work?

Attempt 2.1.22. Observe that from theorem 2.0.13, we know that $|\mathrm{ODD}(2m)|\cdot(2m+1) = |\mathrm{ODD}(2m+1)|$, which resembles our relation $|\mathrm{SQ}(2m)|\cdot(2m+1) = |\mathrm{SQ}(2m+1)|$. Furthermore, we find that $\mathrm{ODD}(n) \subset \mathrm{SQ}(n)$. Hence, this suggests that if Ω is a bijective map from $\mathrm{SQ}(2m)\times[2m+1]$ to $\mathrm{SQ}(2m+1)$, then it may also map $\mathrm{ODD}(2m)\times[2m+1]$ to $\mathrm{ODD}(2m+1)$. We then suspect that $\Psi : \mathrm{ODD}(2m)\times[2m+1] \to \mathrm{ODD}(2m+1)$ defined in the second proof of theorem 2.0.13 may be part of our Ω. Hence, in this attempt, we will try to imitate Ψ for Ω. As it turns out, this approach works perfectly.

I read the proof from [1, Exercise 24, chapter 6] and rewrote the proof as below (with better motivation I believe):

Proof and thoughts. As pointed out in previous, we will try to copy each step in the bijection Ψ from the second proof 2.0.13 and see if it works for Ω.

Consider $(\pi, k) \in \mathrm{SQ}(2m)\times[2m+1]$.

The first step in Ψ is to apply Φ to the permutation, but our permutation $\pi \notin \mathrm{ODD}(2m)$. However, if we break π into two parts, one with even cycles (called even part E) and the other with odd cycles (called odd part O) then the odd part O is completely in $\text{ODD}(2\ell)$ for some ℓ. Hence, we could apply Φ to the odd part of π.

The second step in Ψ is to put $2m+1$ in the k-th gap position in p. Observe that if the k-th gap position is in some even cycle then by adding an extra entry to that cycle, we would mess up the even part E in a way that it's difficult to recover its property 2.1.17. However, if the k-th gap position is in some even odd cycle, we can consider this gap as the gap position for the odd part O and can absolutely apply Ψ to the odd part O and this gap position. In the end, we left the even part E untouched and obtain a new odd part with one more element. The combination of these two parts is a permutation in $\text{SQ}(2m+1)$. Furthermore, observe that $2m+1$ is always in the odd cycle in this case.

We back at the case where the k-gap position is in some even cycle. Again, we wish to leave the structure of even part E unchanged so we cannot add any extra element to the gap position. Instead, we can think such k-th gap position as pointing to the entry x right after that position in that even cycle [1]. In previous case, we put $2m+1$ in an odd cycle so this time we want to put $2m+1$ in a even cycle. This suggests us to replace x with $2m+1$ and we can interpret x as the gap position in the odd part. In particular, if there are $i-1$ entries in the odd part that are less than x then we will mark the i-th gap position in the odd part [2]. It's clear now that we have ourselves an odd part O and a gap position in O so we wish to apply Ψ. However, in Ψ, it's not quite clear which number we will add in the gap position. Note that we want the process to be reversible, i.e. from the new odd part, we want to be able to identify the gap position. This works best if the gap position is the position of the largest entry in the new odd part. Hence, we can do as follow: When running Ψ to the odd part O and the gap position, we label the new entry added as B which will be the largest number in the new odd part. We then shift all elements in odd part O that are larger than X down by one notch [3]. B will automatically replaced with the largest number in the old odd part.

Now, after defining Ω, we will show that Ω is a bijection from $\text{SQ}(2m) \times [2m+1]$ to $\text{SQ}(2m+1)$. Indeed, the above procedure proves that Ω is injective. Hence, it suffices to find Ω^{-1}. Indeed, for each $\pi \in \text{SQ}(2m+1)$, we locate $2m+1$. If $2m+1$ in some even cycle of π then below are the steps:

1. Run the odd part O of π through Ψ^{-1} to identify gap position k in O and at the same time remove the largest entry in O.
2. In canonical form, count how many entries in $\Psi^{-1}(O)$ that are on the left of k-th gap position. Say there are i such entries in $\Psi^{-1}(O)$.
3. Identify $(i+1)$-th smallest entry in $\Psi^{-1}(O)$ and denote such entry as x.
4. Shift entries in $\Psi^{-1}(O)$ that are at least x up one notch. In particular, if in $\Psi^{-1}(O)$, we have $x < a_1 < a_2 < \cdots < a_k$ then $x \to a_1, a_1 \to a_2, \ldots, a_{k-1} \to a_k$ while a_k maps to the largest element in the old odd part O. By doing this, we left out entry x.

[1] the k-th gap position lies between entries x, y in such even cycle so that $y \to x$ for the permutation

[2] Note that x is orginally from an even cycle so if we want to refer to x as the gap position of the odd part O, it's better if we did as above mentioned.

[3] The smallest element in O that is larger than x will be changed to x, the second smallest is changed to the first smallest, etc.

5. Replace $2m+1$ with x to obtain $\Omega^{-1}(\pi) \in \mathrm{SQ}(2m)$ (combines new odd part and new even part). Write π' in canonical form and locate the gap position that lies before x.

If $2m+1$ is in some odd cycle of π then below are the steps:

1. Run the odd part O of π through Ψ^{-1} to identify the gap position in O and at the same time remove $2m+1$ from O.

2. Combine the new odd part with the unchanged even part to get $\Omega^{-1}(\pi)$ and locate the gap position with respect to $\Omega^{-1}(\pi)$.

□

Example 2.1.23. We have $\pi = |(2|1|)(3|)(5|)(6|4|) \in \mathrm{SQ}(6)$ with $|$ refers to the gap position. If the chosen gap position is $k=2$ which is in the even cycle (21) then we select entry 1. We replace 1 with 7 to get $(27)(3)(5)(64)$. The odd part O is $(3)(5)$ so entry 1 refers to the 1st gap position in O since $1<3, 1<5$. We have $\Psi(O,1) = (B)(3)(5)$ and by shifting each entry down by one notch, we get $(5)(1)(3)$. Hence, we obtain $\Omega(\pi,1) = (1)(3)(5)(64)(72)$.

Let's try reverse back using Ω^{-1}. Consider $\pi' = (1)(3)(5)(64)(72)$. We find 7 is in an even cycle so we first run the odd cycle $O = (1)(3)(5)$ through Ψ^{-1} and obtain $(1)(3)$ and 1st gap position of $|(1|)(3|)$. According to the second step, we find that $1,3$ are on the right of 1st gap postiion. In step 3, we find that $x=1$. In step 4, we shift $(1)(3) \to (3)(5)$ up one notch. In step 5, we replace 7 with $x=1$ to obtain $(21)(3)(5)(64)$ in canonical form and obtain 2nd gap position which is located before entry 1. Thus, $\Omega^{-1}(\pi') = ((21)(3)(5)(64), 2) \in \mathrm{SQ}(6) \times [7]$.

Problem 2.1.24. Let k, m and r be positive integers and let $kr = m$. Prove that the number of m-permutations all of whose cycle lengths are divisible by k is

$$\begin{aligned} 1 \cdot 2 \cdots (k-1)(k+1)^2(k+2) \cdots (2k-1)(2k+1)^2(2k+2) \cdots (m-1) \\ = \frac{m!}{k^r r!} \cdot (k+1)(2k+1) \cdots ((r-1)k+1). \end{aligned}$$

Take $k=2$, we get theorem 2.1.19. The idea for this is completely similar to the proof of theorem 2.1.19.

Proof. Consider the canonical form for permutations. Let $\mathrm{DIV}_k(m)$ be number of m-permutations all of whose cycle lengths are divisible by k. For $\pi \in \mathrm{DIV}_k(m)$, then the m-th position in π cannot be m so there are $m-1$ choices for the m-th position. Similarly, there are $m-2$ choices for the $(m-1)$-th position (cannot be m and cannot be the same entry as m-th position),..., $m-k+1$ choices for the $(m-k+2)$-th position. For the $(m-k+1)$-th position, this time m can be in this position so there are $m-(k-1) = m-k+1$ choices for this position (excluding those we've chosen for previous positions). By doing this, we obtain the product $(m-k+1)^2(m-k+2)\cdots(m-1)$. The remaining $m-k$ positions form a permutation in $\mathrm{DIV}_k(m-k)$ so by induction on r, we are done. □

Read more about permutations with roots at [2].

3 Chapter 7: You Shall Not Overcount. The Sieve

Theorem 3.0.1 (Sieve Formula)**.** Let $A_1, \ldots, A_n$ be finite sets. Then

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{j=1}^{n} (-1)^{j-1} \sum_{i_1, i_2, \ldots, i_j} \left|\bigcap_{k=1}^{j} A_{i_k}\right|.$$

The alternating signs is explained by the fact that we have to correct the overcounts.

Here is a formula for the Stirling number of second kind.

Theorem 3.0.2. For all positive integers n and k, the equality

$$S(n,k) = \frac{1}{k!} \sum_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^n.$$

holds.

Note $k! \cdot S(n,k)$ is number of surjections from $[n]$ to $[k]$. A different version of sieve formula:

theo:sieve_2nd_version **Theorem 3.0.3.** Let f and g are functions that are defined on the subsets of $[n]$, and whose range is the set of real numbers. Let us assume that f and g are connected by

$$g(S) = \sum_{T \subseteq S} f(T).$$

Then

$$f(S) = \sum_{T \subseteq S} g(T)(-1)^{|S-T|}.$$

3.1 Exercises

Problem 3.1.1. Let $p = p_1 \cdots p_n$ be a n-permutation. We say that i is a *descent* of p if $p_i > p_{i+1}$. The *descent set* of p is the set of all its decents.

(a) How many 8-permutations have descent set T that is subset of $\{1,4,6\}$?

(b) How many 8-permutations have descent set $\{1,4,6\}$?

(c) How many 8-permutations have descent set $\{1,2,4,5,7\}$?

Proof. (a) If the descent set of p is a subset of $\{1,4,6\}$ then $p_i < p_{i+1}$ for $i \in [8] \setminus \{1,4,6\}$. This follows $p_2 < p_3 < p_4, p_5 < p_6, p_7 < p_8$. A 8-permutation p satisfying the previously mentioned condition will imply that $2,3,5,7,8$ are not descents of p, which means the descent set of p is a subset of $\{1,4,6\}$. Hence, we obtain $\binom{8}{1}\binom{7}{3}\binom{4}{2}\binom{2}{1}$ number of such permutations.

(b) We can use theorem 3.0.3 where $f(S)$ is number of 8-permutations whose descent set is S while $g(S)$ is number of 8-permutations whose descent set is a subset of S. We know that $g(S) = \sum_{T \subseteq S} f(T)$ and from (a), we can calculate $g(S)$ easily. Hence, we can apply theorem 3.0.3 to find $f(\{1,4,6\})$.

(c) The sets looks quite big $\{1,2,4,5,7\}$ for us to apply theorem 3.0.3 as in (b). However, we have the following familiar the bijective map Φ so $\Phi(p_1 \cdots p_n) = (n+1-p_1)(n+1-p_2)\cdots(n+1-p_n)$ (entry in i-th position of $\Phi(p)$ is $n+1-p_i$). With $p \in f(\{1,2,4,5,7\})$, we have $p_1 > p_2 > p_3, p_3 < p_4, p_4 > p_5 > p_6, p_6 < p_7, p_7 > p_8$ Hence, $\Phi(p) = p'_1 p'_2 \cdots p'_8$ with be the reverse of above signs, i.e. $p'_1 < p'_2 < p'_3, p'_3 > p'_4, p'_4 < p'_5 < p'_6, p'_6 > p'_7, p'_7 < p'_8$. This implies $\Phi(p)$ has descent set $\{3,6\}$. Therefore, $f(\{1,2,4,5,7\}) = f(\{3,6\})$.

We have $f(\varnothing) = 1, f(\{3\}) = \binom{8}{3} = 56, f(\{6\}) = \binom{8}{6} = 28, f(\{3,6\}) = \binom{8}{3}\binom{5}{3} = 560$ so $f(\{3,6\}) = 560 - 28 - 56 + 1 = 477$. This implies $f(\{1,2,4,5,7\}) = 477$. □

Remark 3.1.2. Bóna [1, Exercise 11, chapter 7] gives a different bijection for (c). In particular, for a permutation $p_1 \cdots p_8$ we take the *reverse* $p_8 p_7 \cdots p_1$ and the reverse of p will have descent set $\{2,5\}$. This implies $f(\{1,2,4,5,7\}) = f(\{2,5\})$. Interestingly, from our proof for (c), we obtain $f(\{2,5\}) = f(\{3,6\})$.

Problem 3.1.3 (Dual version of theorem 3.0.3)**.** Let h and r be functions that are defined on the subsets of $[n]$, and whose range is the set of real numbers. Assume h and r are connected by $r(S) = \sum_{S \subseteq T} h(T)$. Prove that then

$$h(S) = \sum_{S \subseteq T} r(T)(-1)^{|T-S|}.$$

Proof. For all set V such that $S \subseteq V$, the number of times $h(V)$ will appear once for each set T so $S \subseteq T \subseteq V$ is $(-1)^{|T-S|}$. Hence, the total number of times $h(V)$ will appear is $\sum_{T, S \subseteq T \subseteq V} (-1)^{|T-S|}$. Number of sets T so $|T-S| = i$ is $\binom{|V-S|}{i}$. Therefore, we have

$$\sum_{T, S \subseteq T \subseteq V} (-1)^{|T-S|} = \sum_{i=0}^{|V-S|} (-1)^i \binom{|V-S|}{i} = (1-1)^{|V-S|}.$$

This number is always zero, except when $|V-S| = 0$, i.e. when $V = S$. Thus, the only term on the right-hand side that does not cancel out is $h(S)$, as desired. □

Problem 3.1.4. Let $f(n,k)$ be the number of ways to select a subset of $[n]$, and then select an involution on that subset that has k fixed points. (The empty set has one involution, and that involution has no fixed points). let $g(n) = \sum_{k=0}^{n} f(n,k)(-1)^k$. Prove that $g(n)$ is equal to number of fixed point-free involutions on $[n]$.

Involution here is a permutation p mapping $[k]$ to $[k]$ such that $p^2 = 12 \cdots k$.

Proof. Let $g(n)$ be number of fixed-point-free involutions on $[n]$. We first count number of involutions with k fixed points on a set $S \subseteq [n]$. There are $\binom{|S|}{k}$ ways to choose k elements in S to be a fixed point. The remaining elements form a fixed-point-free involutions on set of $|S| - k$

elements. The number of such involutions is $g(|S|-k)$ so there are a total $\binom{|S|}{k}g(|S|-k)$ desired involutions on S. Thus, we obtain

$$f(n,k) = \sum_{S\subseteq[n]} \binom{|S|}{k} g(|S|-k).$$

We want to show that $g(n) = \sum_{k=0}^{n} f(n,k)(-1)^k$ or

$$g(n) = \sum_{k=0}^{n}(-1)^k \sum_{S\subseteq[n]} \binom{|S|}{k} g(|S|-k). \tag{3.1.1}$$

eqn:sieve_exer_in

Fix $0 \le t,k \le n$, the number of subsets S of $[n]$ with $t+k$ elements is $\binom{n}{t+k}$. This follows $g(t)$ will appear $\binom{t+k}{k}\binom{n}{t+k}$ times for every k. This follows $g(t)$ will appear $\sum_{k=0}^{n}(-1)^k\binom{t+k}{k}\binom{n}{t+k}$ times on the right-hand side of (3.1.1). We have

$$\sum_{k=0}^{n}(-1)^k\binom{t+k}{k}\binom{n}{t+k} = \binom{n}{t}\sum_{k=0}^{n}(-1)^k\binom{n-t}{k} = \binom{n}{t}(1-1)^{n-t}.$$

The above expression is equal to 0 if $t \neq n$ and is equal to 1 if $n = t$, as desired. □

Bijective proof from [1]. It suffices to prove that

$$\sum_{2\mid k} f(n,k) - \sum_{2\nmid k} f(n,k) = g(n).$$

We define a map F. Let $S \subseteq [n]$ and let p be a involution on S. Let M be the largest element of M such that is either a fixed point of p or is not in S. If M is a fixed point of p then remove M from S and remove M from p. The result is a new involution $F(p)$ that has one less fixed point than p. If $M \notin S$ then add M to S and add M as a fixed point to p. The result is a new involution with one more fixed point than p.

Hence, F matches involutions with even number of fixed points to involutions with odd number of fixed points. The only time F is not defined is when M is not defined, which happens when $S = [n]$ and p has no fixed point.

Let E is a set of involutions with even number of fixed points and excluding those involutions on $[n]$ with no fixed points. Let O is a set of involutions with odd number of fixed points then from the above argument, we find F is an injective map from E to O. To show F is surjective, it suffices to find the inverse of F, which is completly similar to F itself. This proves the identity. □

4 Chapter 8: A Function is Worth Many Numbers. Generating Functions.

What's the point of generating functions? The generating function of a sequence contains a lot of information about the sequence, sometimes even more than an exact formula.

4.1 Exercises

> **Problem 4.1.1** (Example 8.14, Exercises 5,6 from [1]). All n soldiers of a military squadron stand in a line. The officer in charge splits the line at several places, forming smaller (non-empty) units. Then he names one person in each unit to be the commander of that unit. Let h_n be the number of ways he can do this.
>
> (a) Find ordinary generating function of $\{h_n\}$ and closed formula for h_n
>
> (b) Show that $h_{n+2} = 3h_{n+1} - h_n$ for $n \geq 1$.
>
> (c) Let F_n be the n-th Fibonacci number. Prove combinatorially that $F_{2n} = h_n$.

Proof. (a) [1, Example 8.14] shows that if $H(x)$ is the ordinary generating function of $\{h_n\}_{n\geq 0}$ then $H(x) = 1 + \frac{x}{1-3x+x^2}$ and also $h_n = \frac{1}{2^n\sqrt{5}}\left[(3+\sqrt{5})^n - (3-\sqrt{5})^n\right]$.

(b) If one use generating function $A(x)$ to find $\{a_n\}$ where $a_{n+2} = 3a_{n+1} - a_n$ for $n \geq 1$ and $a_0 = 1$ then one can obtain $H(x) = A(x)$, which proves the claim.

Here is a different proof: We can view the problem as n points on a line and we want to split the line into smaller units and then choose one point from each unit. Consider a good set of $n+2$ points, i.e. ways to plit $n+2$ points and choose one for each unit. If the last unit has exactly one point which is also the chosen point (for that unit) then we can remove such unit. This gives us a bijection between a good set of $n+2$ points where the last unit has only one point and a good set of $n+1$ points.

If the last unit has more than one points and the last point in the line (which belongs to the last unit) is not the chosen point then we can remove such point out. This gives a bijection to a good set of $n+1$ points.

If the last unit has more than one points and the last point in the line is the chosen point. We remove this last point and choose the second last point (which is also from the last unit). This gives a bijection to a good set of $n+1$ points where the last point is chosen. If a good set of $n+1$ points where the last point is not chosen then from previous paragraph, we find this to be equal to h_n. Hence, by complementary counting, the size of a good set of $n+1$ points where the last point is chosen is $h_{n+1} - h_n$.

In the end, we obtain $h_{n+2} = 3h_{n+1} - h_n$.

(c) One can prove combinatorially (or algebraically) that F_{2n} is number of compositions (i.e. ordered partitions) of $2n-1$ points into parts of size 1 or 2. Say a composition of $2n-1$ has $2k-1$ parts equal to 1 and $n-k$ parts equal to 2. From left to right of the composition, every time we encounter a 2, we replace it with a non-chosen point. If we encounter 1 for the i-th time, if i is odd, then we replace it with a chosen point; if i is even, we replace it with a bar that seperates two units. This will give us k units of n points and each unit has a chosen point. For example, with $n = 8$ and the composition $\alpha = 2+2+1+1+2+1+2+2+1+1 = 15$ which will give us $\circ \circ \bullet | \circ \bullet \circ \circ | \bullet$ where $\circ$ is a non-chosen point and $\bullet$ is a chosen point. $\square$

5 Chapter 9: Dots and Lines. The Origins of Graph Theory

Here are a bunch of definitions:

A diagram made up of points and lines connecting pairs of points is called a *graph*. The dots are the *vertices* and the lines are the *edges* of the graph. Number of edges connecting vertex A is called the *degree* of A. *Loops* are edges that start and end at the same vertex. There can be multiple edges joining the same pair of points. A *simple* graph is a graph with no loops and no multiple edges.

A *walk* is a sequence of alternating vertices and edges so that you can walk through the edges in the order of the sequence. The *length* of a walk is number of edges in the sequence. A walk can be *closed*, where you arrive that the vertex you begin with or it can be *open*.

A *trail* is a walk with no repeated edges. A trail can also be closed (called *circuit*) or open. A *path* is a walk/trail that with no repeated vertices. A path can be closed (called *cycle*, where there is only one repeated vertex, which is the first vertex). If a cycle has k vertices then it has k edges.

From these definitions, we have the following summary:

$$\text{cycle} \subseteq \text{circuit} \subseteq \text{trail} \subseteq \text{walk}; \text{path} \subseteq \text{trail} \subseteq \text{walk}.$$

A *degree sequence* of a graph G is the sequence of non-negative integers whose terms are the degrees of of vertices in G. A degree sequence is usually written in decreasing order. A sequence of non-negative integers is *graphical* if it is the degree sequence of some graph.

Theorem 5.0.1 (Graphical sequence)**.** Let $\Delta = d_1, d_2, \ldots, d_p$ be a sequence S of nonnegative integers with $\Delta = d_1 \geq d_2 \geq \cdots \geq d_p$ and with $p \geq 2, \Delta \geq 1$. Then S is graphical if and only if the sequence

$$d_2 - 1, d_3 - 1, \ldots, d_{\Delta+1} - 1, d_{\Delta+2}, d_{\Delta+3}, \ldots, d_p$$

is graphical.

Essentially we just throw out vertex with degree Δ and edges incident to it.

Proof. The if part is not hard. If we have the graphical sequence $d_2 - 1, \ldots, d_{\Delta+1} - 1, d_{\Delta+2}, \ldots, d_p$ then from the graph whose degree sequence is this sequence, we can add a new vertex of degree Δ by connecting it with Δ vertices of original degree $d_2 - 1, \ldots, d_{\Delta+1} - 1$.

The "only if" part is a bit more difficult. We are given that $\Delta = d_1, d_2, \ldots, d_p$ is a degree sequence that corresponds to vertices $v_1, \ldots, v_p$, respectively. It is not necessary that v_1 is adjacent to $v_2, \ldots, v_{\Delta+1}$ so we cannot always obtain the desired degree sequence by just removing v_1. However, we can try to convert this graph into a different one whose v_1 is adjacent to other Δ vertices with highest degree. We also want our new graph to have the same degree sequence as the original one.

Aiming for these two goals, say if v_1 is not adjacent to a vertex v_i with $2 \leq i \leq \Delta + 1$ then we want our new graph has $v_1 v_i$ as an edge. We also want the degree of v_1 to be the same so an idea is to remove some edge $v_1 v_k$ incident to v_1 and add a new one $v_1 v_i$. Here v_k satisfies $\Delta + 1 < k \leq n$, i.e. it is not one of the Δ vertices other than v_1 with highest degrees. On the other hand, we also want the degrees of v_i, v_k in a new graph to be the same. This suggests us to find a new vertex v_ℓ so v_ℓ is adjacent to v_i but not to v_k. That way we can remove edge $v_\ell v_i$ and add new edge $v_k v_\ell$. Such vertex v_ℓ exists because $\deg v_k = \Delta_k \le \Delta_i = \deg v_i$.

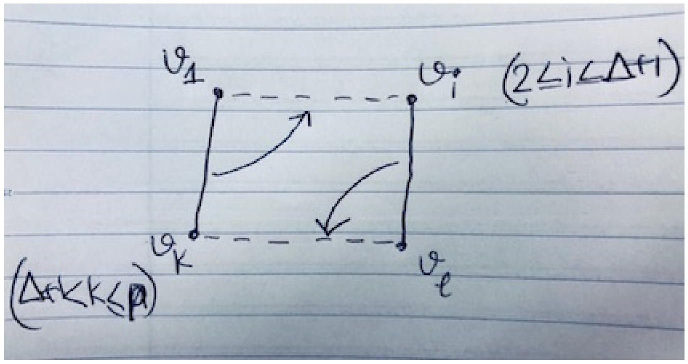


Figure 1: Keep the degrees of v_1, v_i, v_k, v_ℓ unchange. fig:DegreeSeq

The above algorithm creates a graph with vertices $v_1, \ldots, v_p$ with corresponding degrees $\Delta = d_1, \ldots, d_p$ such that v_1 is adjacent to $v_2, \ldots, v_{\Delta+1}$. Hence, we can remove v_1 and edges incident to it to create the graph with desired degree sequence. □

5.1 Eulerian trail and Hamiltonian cycle

A graph is *Eulerian* if it contains a closed trail that includes every edge, such trail is called an *Eulerian trial*. A graph is *semi-Eulerian* if it contains an open trail that includes every edge, such trail is called *semi-Eulerian trail*.

:graphtheory_eulerian **Theorem 5.1.1.** A connected graph is Eulerian iff each vertex has an even degree and is semi-Eulerian if iff it has exactly two vertices of odd degree.

Sketch of proof. $\Longrightarrow$ If G is Eulerian then each vertex has even degree: You starts at some vertex v and every time you enter a new vertex $u \neq v$ from some edge then you must leave v from different edge (since G is a trail). Since the the Eulerian trial covers all edges so the degree of each vertex $u \neq v$ must be even. Since this is a closed trial so the degree of v must be even.

$\Longleftarrow$ If each vertex of G has even degree then G is Eulerian: We starts from a vertex and keep walking until it creates the first circuit (closed trail) C_1. This is possible because: G is finite; every vertex has even degree, which means every time you enter one vertex then you can get out of it with different edge.

If C_1 is G then we are done. If not, there exists vertex A not in G and since G is connected (as a closed trail), there exists vertex v in C_1 adjacent to some $u \notin C_1$. This means we can get out of this circuit C_1 through v.

Omit all edges of C_1 from G and we obtain a graph with all vertices having even degree. Starting at the mentioned vertex v from C_1 and take another closed trail C_2. We can then unite C_1 and C_2 into one closed trail by: start walking at C_1, stops at v then walk through C_2 then complete the trail by using remaining edges of C_1.

Thus, $C_1 \cup C_2$ is a closed trail. We proceed similar from here.

For semi-Eulerian graph, add an extra edge between two vertices with odd degree then we obtain an Eulerian trail. If we delete the added edge we will get a semi-Eulerian trail (see the trail as a sequence of vertices, and see how deleting an edge between consecutive vertices changes to an semi-Eulerian trail).

(Input diagram) □

A cycle that includes all vertices of a graph is called a *Hamiltonian cycle*, whereas a path that includes all vertices of a graph is called a *Hamiltonian path*.

> Eulerian trail is a closed trail that includes all edges in G. Hamiltonian cycle is a cycle (in some sense "closed path") that includes all vertices (called *spanning subgraph*) of G.

Theorem 5.1.2. If there exists a subset S of the vertices of a graph G such that $G \setminus S$ has more than $|S|$ components then G has no Hamiltonian cycle.

Proof. We proceed by contradiction (as this will give us a Hamiltonian cycle to work with). Assume the contrary, G has one Hamiltonian cycle H. Let the components of $G \setminus S$ be $G_1, \ldots, G_t$ where $t > |S|$.

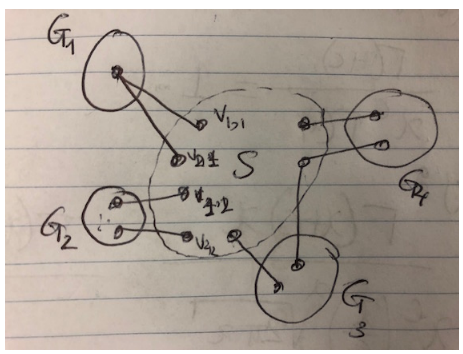


Figure 2: S and connected components of $G \setminus S$ fig:hamiltoniancy

Since G has no edges joining G_i and G_j for $i \neq j$ and G needs to be connected to have a Hamiltonian cycle so there are edges joining each G_i to S. In fact, there are at least two edges joining each G_i with S whose endpoints in S are different from each other. Indeed, consider the Hamiltonian cycle where you starts at some v_i from G_i (you can start anywhere you want for the Hamiltonian cycle because it is a cycle) then you need to go to S first before returning to v_i again. Say that your first meeting with S is with vertex $u_1 \in S$ and your last meeting with S is with $u_2 \in S$. These two u_1, u_2 need to be distinct, otherwise if $u = u_1 = u_2$ then that means you've gone through u two times [4] while u is not the starting point, a contradiction.

For each $i = 1, \ldots, t$, let $v_{1,i}, v_{2,i}$ be the two distinct vertices in S that corresponds to G_i as from the previous arguments. Note that for $i \neq j$ $v_{1,i}, v_{1,j}$ or $v_{1,i}, v_{2,j}$ need not to be distinct. Hence, we obtain a multisubset $T = \{v_{1,i}, v_{2,i} : 1 \leq i \leq t\}$ of set of vertices of S. Note that each vertex in

[4] there is a degenerate case where you return to G_i to v_i right away after visiting u, which means $t = |S| = 1$, which is also a contradiction

G must be incident to exactly two edges in the Hamiltonian cycle H. This follows each element s in T appear in T for at most two times (otherwise s is adjacent to at least three edges in H). Therefore, T must have at least t distinct elements. Thus, $t \le |S|$, which is a contradiction. □

Remark 5.1.3. The converse of this is not true. Consider Petersen graph.

Theorem 5.1.4 (Dirac's theorem)**.** Let G be a graph with $n \ge 3$ vertices. If every vertex in G has degree at least $n/2$ then G has a Hamiltonian cycle.

Proof. Suppose G does not have a Hamiltonian cycle.

Consider the extreme case of the graph: We add new edge as long as we can without creating a Hamiltonian cycle. Our new graph G' has all vertices with degree at least $n/2$. G' has no Hamiltonian cycle but adding any extra edge to G' will create one.

Consider vertices x, y in G that are not adjacent to each other (there must exists such pair (x, y) because G is not Hamiltonian). Since adding edge xy to G will create a Hamiltonian cycle so there is a Hamiltonian path in G that starts with x and ends with y. Let such path be $x = v_1, v_2, \ldots, v_{n-1}, v_n = y$.

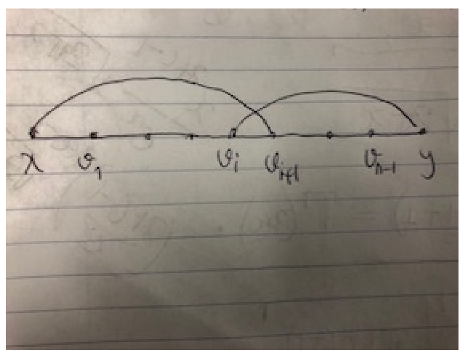


Figure 3: Path from x to y fig:hamiltoniancy

Since x is adjacent to at least $n/2$ vertices v_i for $2 \le i \le n-1$ so there are at least $n/2 - 1$ vertices v_i $(2 \le i \le n-1)$ such that xv_{i+1} is edge in G. Let the set of such vertices as X. There are also $n/2$ vertices v_j $(2 \le j \le n-1)$ that are adjacent to y, let the set of such vertices as Y. This follows $|X| + |Y| \ge n - 1$ but $|X \cup Y| \le n - 2$ so $|X \cap Y| \ge 1$, i.e. there exists v_i $(2 \le i \le n-2)$ such that $v_i y, v_{i+1} x$ are edges of G. Then $xv_2v_3 \cdots v_i y v_{n-1} \cdots v_{i+1} v_i$ is a Hamiltonian cycle, a contradiction. □

Example 5.1.5 (Exercise 14, §6 from [1])**.** Petersen diagram has no Hamiltonian cycle.

Indeed, call the five edges joining an outer vertex to an inner vertex sticks. Any Hamiltonian cycle would have to contain even number of sticks (imagine you start at an outer vertex then in order to get back to that vertex, you go in and out the region of inner vertices for even number of times, which results in even number of sticks). Hence, there can be 2 sticks or 4 sticks. Each case will give a contradiction.

5.2 Directed graphs

Each edge in the direct graph has a unique direction so this needs to be taken into account for the definition of trails, closed trails, paths for directed graphs.

For Eulerian trail, the condition in theorem 5.1.1 no longer holds. For example, if no edge starts at a given vertex then there will be no Eulerian trail in that graph.

A directed graph G is *strongly connected* if for all vertices u and v of G, there is a directed path from u to v. The *in-degree* of a vertex of a directed graph is number of edges that end at that vertex. The *out-degree* of a vertex is number of edges that start at that vertex. A directed graph G is *balanced* if for each vertex $v \in G$, the in-degree of v equals the out-degree of v.

Theorem 5.2.1. A directed graph G has a closed Eulerian trail if and only if it is balanced and strongly connected.

Proof. We prove the two conditions are necessary. Indeed, a closed trail of G visits each vertex as many times it enters that vertex, which shows that G is balanced. G has a closed trial implies that there is a trail from any two vertices in G, which impies G is strongly connected.

For the converse that the conditions are sufficient, copy theorem 5.1.1 and replace edges by directed edges. □

A simple undirected graph is called *complete* if there is an edge between any pair of distinct vertices. A complete graph on n vertices has $\binom{n}{2}$ edges. If we direct each edge of a complete graph then the resulting directed graph is a *tournament*. The reason for the name is that each edge from i to j could mean i beats j in a tournament.

Theorem 5.2.2. All tournaments have a Hamiltonian path.

Proof. Induction on number of vertices. □

For determining whether a tournament is a Hamiltonian cycle:

Theorem 5.2.3. A tournament T has a harmiltonian cycle if and only if it is strongly connected.

Proof. If T has Hamiltonian cycle then for any two vertices in T, there exists a directed path that goes from one to the other. This follows T is strongly connected.

Conversely, if T is strongly connected. First we show that there exists a directed cycle in T. Indeed, if not then for any $x, y, z \in V(T)$, if $xy, yz \in T$ then $xz \in T$. This follows T is a *transitive tournament*, i.e. the vertices of T can be listed such that $ij \in E(T)$ iff j is on the right of i in the list. However, such tournament is not strongly connected as you can't go from the last vertex in the list to the first vertex in the list. Thus T does have a cycle.

Consider the cycle $C = y_1 \cdots y_k$ of T with the largest length. Assume that C is not a Hamiltonian cycle then there exists a vertex $x \notin C$. Since T is strongly connected so there exists, WLOG say $y_1 \in C$ such that $y_1 x \in E(T)$ (otherwise you can't go from y_1 to x). If $xy_2 \in E(T)$ then $y_1 x y_2 \cdots y_k$ is a bigger cycle than C, a contradiction. Thus, we must have $y_2 x \in E(T)$ and similarly, $y_i x \in E(T)$ for all $1 \le i \le k$.

Let Z be set of vertices in $T \setminus C$ such that $y_1 z \in E(T)$ for each $z \in Z$. This follows $y_i z \in E(T)$ for every $1 \le i \le k$ and every $z \in Z$. Since there is a directed path from z to y_1 so there exists

$t \notin Z \cup C$ such that $zt \in E(T)$. Since $t \notin C \cup Z$ so $ty_1 \in Z$. In the end, $zty_1y_2 \cdots y_k$ is a bigger cycle than C, a contradiction.

Thus, C must be an Hamiltonian cycle.

(Include diagram) □

5.3 The notion of Isomorphisms

Two graphs are isomorphic if they are identical as unlabeled graph:

Definition 5.3.1. We say that graphs G and H are isomorphic if there is a bijection f from the vertex set of G onto that of H so that the number of edges between any pairs of vertices u, v in G is equal to number of edges between vertices $f(u)$ and $f(v)$ of H. The bijection f is called an isomorphism.

5.4 Exercises

Problem 5.4.1. Let G be a loopless undirected graph. Prove that the edges of G can be directed so that no directed cycle is formed.

Proof. Label the vertices as $1, 2, \ldots, |G|$. If ij is an edge in G where $i < j$ then we draw arrow from i to j. The labels increase along any directed graph, so no directed cycle exists. □

Problem 5.4.2. Let G be a simple graph on 10 vertices and 28 edges. Prove that G contains a cycle of length 4.

Proof. Consider a general case where G has n vertices $v_1, \ldots, v_n$ and e edges. For each $i \in [n]$, denote V_i as set of vertices that are adjacent to v_i. If G has no cycle of length 4 then for any $i \neq j$, $|V_i \cap V_j| \le 1$. Therefore, $\sum |V_i \cap V_j| \le \binom{n}{2}$.

On the other hand, $\sum |V_i \cap V_j| = \sum \binom{|V_i|}{2}$. We know $\sum |V_i| = 2e$ so $\sum \binom{|V_i|}{2} \ge e\left(\frac{2e}{n} - 1\right)$. This establishes the inequality $n(n-1) \ge 2e\left(\frac{2e}{n} - 1\right)$. Check with $n = 10, e = 28$, we find that the graph G must contain cycle of length 4. □

Problem 5.4.3 (Exercise 13 from [1])**.** The previous exercise defines a regular graph as a simple graph in which each vertex has the same number of neighbors. Is it true that in such graph, each vertex will have the same number of second neighbors? (The vertex X is a second neighbor of a vertex Y if XY is not an edge, and there is a path of length 2 joining X and Y).

Proof. We predict the answer to be no. Why no? Say the degree of each vertex is 3 and we start from one vertex v and draw out its neighbors and second neighbors. There are possible cases:

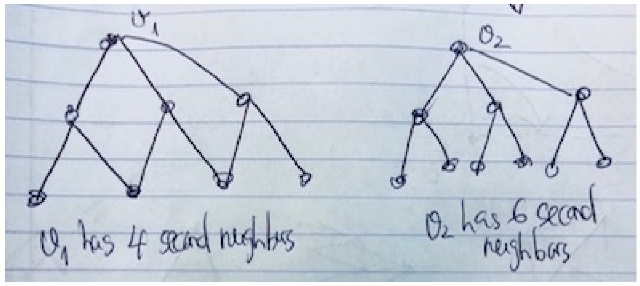


Figure 4: Possible cases for different number of second neighbors fig:exer13chap6_

With this idea in mind, we have the following counterexample:

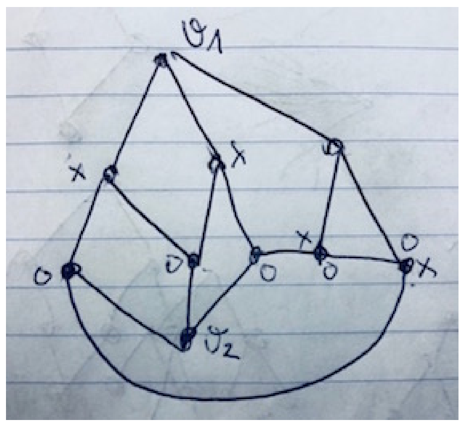


Figure 5: v_1 has five second neighbors, labeled $\circ$, while v_2 has four second neighbors, labeled $\times$ fig:exer13chap6_

We are done. □

Problem 5.4.4. The degree sequence of a graph G forms a partition of $2|E(G)|$. This partition is never self-conjugate.

Proof. Because if $d_1, \dots, d_n$ is a degree sequence of G then the conjugate of this will have k parts, where k is number of vertices with positive degree. On the other hand, the partition has at most $k-1$ parts as $d_1 \leq k-1$. □

Problem 5.4.5. How many automorphisms does the graph shown in [1, Figure 9.11, §9] have?

Proof. Consider a bijection f from set of vertices to itself. There are 8 choices for $f(A)$. After finding $f(A)$, let S be the set of vertices that are adjacent to $f(A)$ then $S = \{f(D), f(E), f(B)\}$. Hence there are 3! ways to map D, E, F to one of vertices in S. After finding $f(A), f(D), f(E)$, we can uniquely determine $f(H)$ as the vertex that is adjacent to $f(D), f(E)$ but is different from $f(A)$. Similarly, there is one way to choose $f(F)$ and one way to choose $f(C)$. After we found out about $f(C), f(H), f(F)$, then $f(G)$ is uniquely determined. Thus, there are $8 \cdot 3! = 48$ isomorphisms. □

Problem 5.4.6. How many automorphisms does the graph in figure [1, Figure 9.12, §9] have?

Proof. Consider such isomorphism f.

Since each vertex has the same role (for example, each of them are adjacent to 4 other vertices, that four vertices form a cycle, that four vertices are adjacent to another vertex different from the initial one). Hence, there are 6 ways to choose $f(F)$. Since there exists exactly one vertex not adjacent to $f(F)$ so there is one way to choose $f(E)$.

Since $f(A), f(B), f(C), f(D)$ are vertices adjacent to $f(F)$ and they form a cycle so there are 4 ways to choose $f(A)$ (from vertices adjacent to $f(F)$. Since $f(D), f(B)$ adjacent to $f(A)$ so there are 2 ways to choose $f(D), f(B)$ and finally, one way to choose $f(C)$.

Thus, we obtain $6 \cdot 4 \cdot 2 = 48$ ways. □

6 Chapter 10: Staying connected. Trees

6.1 Minimally connected graphs

theo:tree_definitions **Theorem 6.1.1** (Equivalent definitions for a tree)**.** Let G be a connected simple graph on n vertices. Then the following are equivalent:

1. G is minimally connected, that is, if we remove any edge of G, then the obtained graph G' will not be connected.

2. G does not contain a cycle.

A simple connected graph satisfying one of three above conditions is called a *tree*.

Sketch. (1) $\implies$ (2): If G is minimally connected while still having a cycle C, one can remove an edge of C and obtain a new connected graph. This follows a contradiction. Thus, G does not contain a cycle.

$\sim$ (1) $\implies \sim$ (2): (prove by contrapositive). Assume G is not minimally connected. This follows there exists an edge e from A to B such that $G' = G \setminus \{e\}$ is connected. This follows there exists a path P from A to B in G'. This implies $P \cup \{e\}$ is a cycle in G. □

corol:tree_UniquePath **Corollary 6.1.2.** A connected graph G is a tree if and only if for each pair of vertices (x, y), there is exactly one path joining x and y.

Proof. If for each pair of vertices (x, y), there is exactly one path from x to y then G is minimally connected (proof is similar to theorem 6.1.1 at the part (1) $\implies$ (2).

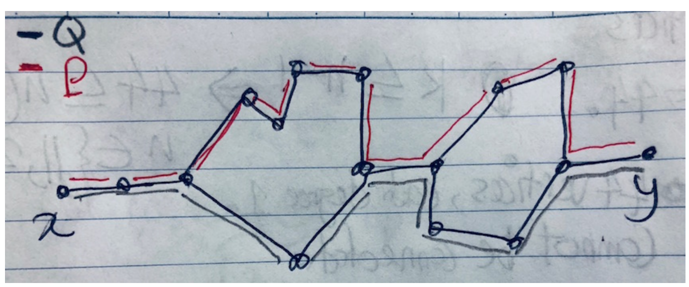


Figure 6: Cycles create from symmetric difference of two paths P, Q fig:Tree_UniquePa

If G is a tree, but there are two paths P and Q joining vertices x and y. Take the *symmetric difference* of P and Q, that is, the edges that are part of exactly one of P and Q. This symmetric difference is a union of cycles. □

aph_tree_edgecounting **Theorem 6.1.3.** All trees on n vertices have $n - 1$ edges. Conversely, all connected graphs on n vertices with exactly $n - 1$ edges are trees.

Vertices of tree that have degree 1 are called *leaves*.

Proof. First, we show that for $n \geq 2$ then any tree T with n vertices has at least two leaves. Indeed, take any path p with maximum length in T then two endpoints of p must be leaves.

Back to the problem, we use induction on n. The case $n = 1$ is trivial. For a tree T with $n+1$ vertices, delete a leaf and its edge e of T to obtain new tree T' with n vertices. According to inductive hypothesis, T' has $n-1$ edges so $T = T' \cup \{e\}$ has n edges. □

A *forest* is a graph in which each connected component is a tree.

ee_forest_edgecounting **Corollary 6.1.4.** Let F be a forest on n vertices with k connected components. Then F has $n-k$ edges.

We now interest in counting number of trees on n vertices where the vertices are distinguishable, i.e. counting all trees with vertex set $[n]$.

o:tree_counting_cayley **Theorem 6.1.5** (Cayley)**.** For any positive integer n, the number of all trees with vertex set $[n]$ is $A_n = n^{n-2}$.

By André Joyal. Take A_n trees on $[n]$, and in each of them, choose two not-necessarily distinct vertices, and call one of them Start, and the other one End. We can do this n^2 ways for each tree. Call the $n^2 A_n$ objects obtained *doubly rooted trees*.

We will show that number of doubly rooted trees on $[n]$ is n^n by constructing a bijection from the set of all functions from $[n]$ to $[n]$ and that of doubly rooted trees on $[n]$.

Let f be a function from $[n]$ to $[n]$. Let C be subsets of elements $x \in [n]$ which are part of a cycle under the action of f, in other words, $x \in C$ if $f^i(x) = x$ for some positive integer i. Let $C = \{c_1 < \ldots < c_k\}$. Let $d_i = f(c_i)$. Note that $\{d_i\} = C$. We construct a path $S = d_1 \cdots d_k$ where we mark d_1 as Start and d_k as End. If $j \notin C$ then joint vertex j with vertex $f(j)$.

We show that we've constructed a tree on $[n]$ from f. Indeed, the vertex set of the new graph T is indeed $[n]$ as we first put in vertices from C then we put vertices not in C in. The graph T is connected because every vertex is connected to the path $d_1 \cdots d_k$.

Observe that for any $i \in [n]$ then i is adjacent to $f(i)$ and (if exists, one or more) $f^{-1}(i)$'s. This follows if T has a cycle C' then its vertex set must be $\{f^i(x) : 1 \leq i \leq \ell\}$ for some $x \in C, k \in \mathbb{Z}_{\geq 1}$. Since any $x \in C$ lies on the path S so there does not exist such cycle. This follows T is cycle-free. Thus, T is a doubly rooted tree on $[n]$.

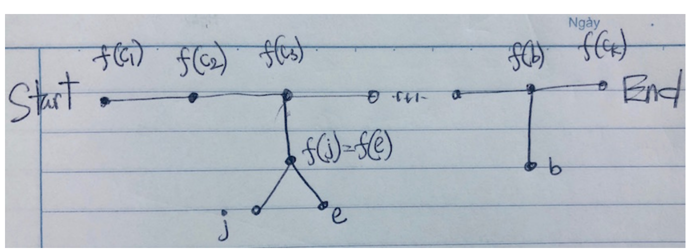


Figure 7: Assign vertices labels for doubly rooted tree fig:CayleyProof_I

corol:tree_UniquePath

Conversely, for any doubly rooted tree on $[n]$, from corollary 6.1.2, there exists a unique path from Start to End consisting of vertex set C. Define f on C such that the image of the smallest

i-th element in C is the i-th vertex on the Start-End path from Start. For $j \notin C$, i.e. for vertex j not on the Start-End path, $f(j)$ is the first neighbor of j on the unique path from j to the Start-End path. $\square$

We can define *rooted trees* to be trees with one vertex called the root. So the number of rooted trees on $[n]$ is n^{n-1}. A *rooted forest* is a forest in which each component is a rooted tree.

Second proof. This is from [1, Exercise 5, §10]. Let T be a tree on $[n]$, with $n \geq 3$. Cur of the leaf of T that has the smallest label, and write down its single neighbor. Then continue this same procedure on the remaining tree until there are only two vertices (and one edge) left. This procedure results in a sequence of elements of $[n]$ that has length $n-2$ (repetition allowed), called the *Prűfer sequence* of T.

We show that this algorithm defines a bijection from the set of all trees on $[n]$ onto the set S_n of sequences of length $n-2$ with elements from $[n]$. Indeed, it suffices to construct a tree from each sequence in S.

We've already defined an injecttion from set of trees on $[n]$ and set of Prű sequence on $[n]$. It suffices to construct an injection from set of Prű sequence on $[n]$ to set of trees on $[n]$. Indeed, consider a sequence $S = a_1 \cdots a_{n-2}$ whose elements in $[n]$. We construct a tree T from this sequence as follow:

1. Let b be the largest number in $[n]$ that does not appear in S. First we add edge ba_{n-2} to T so T now only has b, a_{n-2} as vertices and ba_{n-2} as the only edge.

2. Inductively with i from $n-2$ to 1: Consider a_i, if a_{i-1} already appear in the tree or if $i = 1$ then let b be the largest number in $[n]$ that is not currently in the tree, we add b to the tree by connecting b with a_i.

   If a_{i-1} does not appear in the current tree then add a_{i-1} to the tree by connecting a_{i-1} with a_i.

The resulting graph T is a graph of n vertices on $[n]$ because no repetition of labels is guaranteed in each step and since we add a new vertex for every a_i, we get a total of n vertices with pairwise different labels from $[n]$.

The graph is connected because when we add new vertex to the graph, we make sure to connect it with current graph. The graph has exactly $n-1$ edges because in each step of considering a_i, we add exactly one edge to the graph, plus the original edge in step 1, we obtain a total of $n-1$ edges.

From this and theorem 6.1.3, we find that our graph is a tree on $[n]$. This mapping is obviously an injection.

Thus, we've constructed a bijection. Furthermore, there are n^{n-2} sequences of length $n-2$ with elements in $[n]$ so this proves that number of trees on $[n]$ is n^{n-2}. $\square$

Corollary 6.1.6. For all positive integer n, the number of rooted forests on $[n]$ is $(n+1)^{n-1}$

Proof. Take a rooted forest on $[n]$, and join all roots to the new vertex $n+1$ by an edge. This gives us an unrooted tree on $[n+1]$. Conversely, given a tree on $[n+1]$, make all vertices adjacent to $n+1$ as roots then remove $n+1$ to obtain a rooted forest. Thus, there is a bijection between set of rooted forests on $[n]$ and set of trees on $[n+1]$. Previous theorem tells us the answer of $(n+1)^{n-1}$. $\square$

6.2 Mimimum-weight spanning trees. Kruskal's greedy algorithm

A *spanning tree* is a spanning subgraph that is a tree. Any connected graph G has at least one spanning tree. One can always omit edges of G to obtain a minimally connected graph, which is a tree. In general, a connected graph has many spanning trees. Cayley's theorem 6.1.5 shows that K_n has n^{n-2} spanning trees.

Let G be a connected simple graph. Let $w : E(G) \to \mathbb{R}^+$ be a function. We aim to find a spanning tree T of G such that $\sum_{e \in T} w(e)$ is minimal.

The function w is called the *weight function* or *cost function* of G, and $w(e)$ is called the *weight* or *cost* of e, while $\sum_{e \in T} w(e)$ is called the weight of T.

The spanning tree T can be constructed greedily: Take the edge with the smallest weight and put it in T. Second, look for the edge that has the smallest weight among those not in T, and add it to T. For the i-th step, we need to add an edge e_i to T such that:

(a) The edge e_i is not yet in T,

(b) If we add edge e_i in T, the obtained graph does not contain a cycle,

(c) The weight of e_i is minimal among all edges that have properties (i) and (ii).

We show that T is a spanning tree of G after exactly $n-1$ steps. Note that at every step $i \le n-1$, T is a forest so the algorithm must terminate after at most $n-1$ steps (since if more than $n-1$ steps means T is a forest with more than $n-1$ edges, a contradiction). On the other hand, we can proceed the algorithm up until $n-1$ steps (i.e. until T has $n-1$ edges). Indeed, if T has at most $n-2$ edges and T is a graph with n vertices so T is not connected, while G is connected, so there exists an edge in G that joins two connected components of T and therefore, can be added to T.

We proved that the algorithm stops after exactly $n-1$ steps. After this, T is a forest with $n-1$ edges so T is a tree using corollary 6.1.4. Thus, the greedy algorithm can find one spanning tree T of connected graph G. The next question is whether this spanning tree T a minimum-weight spanning tree.

lem:tree_forest **Lemma 6.2.1.** Let F and F' be two forests on the same vertex set V, and let F have less edges than F'. Then F' has an edge e that can be added to F such that the obtained graph $F \cup \{e\}$ is still a forest.

Proof. Assume that there is no such edge e in F' then adding any edge of F' to F would create a cycle in F. This follows any edge of F' is between vertices from the same connected component of F. Hence, if a connected component of F (which is a tree) has k vertices then edges of F' in this connected component is at most $k-1$ edges. Summing up, we obtain that number of edges in F' is at most number of edges in F, a contradiction. $\square$

Theorem 6.2.2. The Kruskal's Greedy Algorithm always find the minimum-weight spanning tree.

Proof. Assume the greedy algorithm gives us the spanning tree T, whereas our graph G has a spanning tree H whose total weight is less than that of T. Let $h_1, \dots, h_{n-1}$ be edges of H so that $w(h_1) \le w(h_2) \le \dots w(h_{n-1})$ holds. Let $t_1, \dots, t_{n-1}$ be edges of T such that $w(t_1) \le w(t_2) \le \dots \le w(t_{n-1})$ holds. Note that $t_1, \dots, t_{n-1}$ is also the steps for the algorithm to obtain T (if in different order would contradict the (c) requirement for the algorithm).

Let i be the step at which H beats T, i.e. i as the smallest integer such that $\sum_{j=1}^{i} w(h_j) < \sum_{j=1}^{i} w(t_j)$. Such i exists since H will eventually beats T after $n-1$ steps. Also note that $i > 1$ since $w(t_1)$ is the minimal among all edge-weights of G according to the algorithm. Due to the minimality of i, we have

$$\sum_{j=1}^{i} w(h_j) < \sum_{j=1}^{i} w(t_j), \sum_{j=1}^{i-1} w(h_j) \ge \sum_{i=1}^{i-1} w(t_j).$$

This follows $w(h_i) < w(t_i)$. We deduce the contradiction by showing that at i-th step, the algorithm cannot add t_i to T. Let T_{i-1} be the forest that the algorithm produced at step $i-1$, i.e. it contains all edges $t_1, \dots, t_{i-1}$. Define H_i similarly. Applying lemma 6.2.1 to H_i and T_{i-1}, there exists $h_j \in H_i$ $(j \le i)$ such that $T_{i-1} \cup \{h_j\}$ is a forest. On the other hand $T_i = T_{i-1} \cup \{t_i\}$ and $w(h_j) \le w(h_i) < w(t_i)$ so this implies that the algorithm cannot add t_i at step i since t_i does not give minimal weight among the edges that could be added to T_{i-1} without forming a cycle. □

6.3 Graphs and Matrices

6.3.1 Adjacency matrices of graphs

Definition 6.3.1. Let G be an undirected graph on n labeled vertices, and define an $n \times n$ by $A = A_G$ by setting $A_{i,j}$ equal to number of edges between vertices i and j. Then A is called the *adjacency matrix* of G.

If G is directed, then we can define its adjacency matrix by setting $A_{i,j}$ equal to number of edges *from* i to j. Thus, the adjacency matrix of a directed graph is not necessarily symmetric, while that of an undirected graph is.

Theorem 6.3.2 (Power of adjacency matrix)**.** Let G be a graph on n labeled vertices, let A be its adjacency matrix, and let k be a positive integer. Then $A_{i,j}^k$ is equal to number of walks from i to j that are of length k.

Sketch. Induction on k. □

Theorem 6.3.3 (Connectivity testing with adjacency matrix)**.** Let G be a simple graph on n vertices, and let A be the adjacency matrix of G. Then G is connected if and only if $(I+A)^{n-1}$ consists of strictly positive entries.

Sketch. There is a path from i to j iff there exists $k \le n-1$ such that $A_{i,j}^k > 0$. □

6.3.2 Number of spanning trees of a graph

H is a spanning tree of a directed graph G if H is a subgraph of G and when removing the orientations of all edges to get H_1 and G_1, H_1 is a spanning tree of G_1.

Definition 6.3.4 (Incidence matrix)**.** Let G be a directed graph without loops. Let $\{v_1, \dots, v_n\}$ denote the vertices of G, and let $\{e_1, \dots, e_m\}$ denote the edges of G. Then the incidence matrix of G is the $n \times m$ matrix A defined by

- $a_{i,j} = 1$ if v_i is the head of e_j,
- $a_{i,j} = -1$ if v_i is the tail of e_j, and
- $a_{i,j} = 0$ otherwise.

Theorem 6.3.5. Let G be a directed graph without loops, and let A be the incidence matrix of G. Remove any row from A, and let A_0 be the remaining matrix. Then the number of spanning trees in G is $\det A_0 A_0^T$.

Proof. Denote the incidence matrix A of G as in definition 6.3.4. WLOG, assume that the last row (whose associated vertex is v_n) of A was omitted so A_0 is a $(n-1) \times k$ matrix. Denote $\binom{[k]}{n-1}$ as set of all subsets of size $n-1$ of $[k]$. Then for $S \in \binom{[k]}{n-1}$, let A_S be a $(n-1) \times (n-1)$ submatrix of A_0 whose columns are columns of A_0 at indices from S. By the Cauchy-Binet formula, we have

$$\det(A_0 A_0^T) = \sum_{S \in \binom{[k]}{n-1}} (\det A_S)(\det A_S^T) = \sum_{S \in \binom{[k]}{n-1}} (\det A_S)^2.$$

Let B_S be the subgraph of G whose edges from columns of A_0 at indices from S. Note that for each $S \in \binom{[k]}{n-1}$, B_S has n vertices and $n-1$ edges corresponding to $n-1$ columns of A_S. We show that $|\det A_S| = 1$ if and only if B_S is a spanning tree and $\det A_S = 0$ otherwise. With this and the above formula, we can get what we want.

To prove the above claim, we induct on n:

(a) If there is a vertex $v_i (i \neq n)$ of degree 1 in B_S. Then the i-th row of A_S[5] contains exactly one nonzero element, namely 1 or -1 at j-th column. Expanding A_S along this row, then $|\det(A_S)| = |\det(C_S)|$ where C_S is $(n-2) \times (n-2)$ submatrix of A_S by deleting i-th row and j-th column. Also note that B_S is a spanning tree iff $B_S \setminus \{v_i\}$ is a spanning tree, whose edge-set corresponds to columns of C_S. By inductive hypothesis, we are done.

(b) If B_S has no vertex of degree 1 (except possibly v_n). Then B_S is not a spanning tree [6] and since it has $n-1$ edges so there must be one with degree zero. If such vertex is not v_n then the matrix A_S has a zero row, which means $\det A_S = 0$. If such vertex is v_n then all edges have endpoints in $\{v_1, \dots, v_{n-1}\}$. This follows every column of A_S (corresponding to an edge in B_S) contains one 1 and one -1 as each edge has a head and a tail in $\{v_1, \dots, v_{n-1}\}$. Hence, the sum of rows of A_S is 0 so the rows is linearly dependent, so $\det A_S = 0$. Thus, in this case, we know that B_S is not a spanning tree and that $\det A_S = 0$.

[5]note that $i \neq n$ so such row exists in A_S as A_S has $n-1$ rows correspondings to vertices $v_1, \dots, v_{n-1}$

[6]a spanning tree needs at least two vertices of degree 1 according to proof of theorem 6.1.3 while B_S has at most one such vertex

This proves the claim. □

What about undirected graph? Based on the definition of spanning tree for directed graph and the above theorem, for a simple undirected graph G, we can just add the orientations for the edges to create an directed graph G' from G and use the above theorem. Using this idea, we have the following theorem:

Theorem 6.3.6 (Matrix-tree theorem)**.** Let U be a simple undirected graph. Let $\{v_1, \dots, v_n\}$ be the vertices of U. Define $(n-1)\times(n-1)$ matrix L_0 by

$$L_{i,j} = \begin{cases} \deg v_i & i = j, \\ -1 & i \neq j, v_i \text{ and } v_j \text{ are adjacent}, \\ 0 & \text{otherwise}, \end{cases}$$

where $1 \le i, j \le n-1$. Then U has exactly $\det L_0$ spanning trees.

Proof. We turn U into a directed graph G by replacing each edge of U by a **pair** of directed edges, one edge going in each direction.

Let A_0 be the indicence matrix of G. We show that $A_0A_0^T = 2L_0$. Indeed, the (i,j) position of $A_0A_0^T$ is the scalar product of the i-th row and the j-th row of A_0.

(a) If $i = j$, then every edge in G that starts or ends at v_i contributes 1 to $L_{i,i}$. Hence, the (i,i)-th entry of $A_0A_0^T$ is the degree of v_i in G or twice the degree of v_i in U.

(b) If $i \neq j$, then every edge that starts at v_i, ends at v_j or starts at v_j, ends at v_i contributes -1 to $L_{i,j}$. Since U is a simple graph, there is either 0 or 1 edge joining v_i and v_j in G. This follows the (i,j)-th entry of $A_0A_0^T$ is -2 if v_iv_j is an edge in G and 0 otherwise.

This proves that $A_0A_0^T = 2L_0$. This follows $2^{n-1}\det L_0 = \det A_0A_0^T$. Since for each spanning tree in U, one can create 2^{n-1} spanning trees for G by orienting its $n-1$ edges. Therefore, our statement immediately follows from theorem 6.3.5. □

Example 6.3.7. We reprove Cayley's theorem 6.1.5 about K_n having n^{n-2} spanning trees by using theorem 6.3.6.

The matrix L_0 associated to K_n is

$$\begin{pmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & n-1 & -1 & \cdots & -1 \\ -1 & -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & n-1 \end{pmatrix}$$

Example 6.3.8. The matrix L_0 associated to the *complete bipartite graph* $K_{m,n}$ is

$$\begin{pmatrix} n & \cdots & 0 & -1 & \cdots & -1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & n & -1 & \cdots & -1 \\ -1 & \cdots & -1 & m & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -1 & \cdots & -1 & 0 & \cdots & m \end{pmatrix}$$

That is, the first m rows look "similar", the last $n-1$ rows look "similar". The same is true for columns. The determinant of this matrix is $n^{m-1}m^{n-1}$, which is number of spanning trees of $K_{m,n}$.

Theorem 6.3.9 (Eigenvalue-version of Matrix-Tree theorem)**.** Let U be the graph as in theorem 6.3.6 and L be defined the same way as L_0 in theorem 6.3.6, except that L is an $n \times n$ matrix. Denote $\lambda_1, \dots, \lambda_n$the eigenvalues of L, with $\lambda_n = 0$. Then the number of spanning trees of U is $\frac{1}{n}\lambda_1 \cdots \lambda_{n-1}$.

Remark 6.3.10. 0 is always an eigenvalue of L because sum of entries in each column of L is 0 so the rows of L add up to zero row, which means they are linear independent.

Example 6.3.11. If U is a *regular* graph where each vertex in U has degree d, then $dI - A = L$ where A is the adjacency matrix of U the L is the matrix defines in theorem 6.3.9. This follows if $\alpha_1, \dots, \alpha_n$ is the eigenvalues of A then $d - \alpha_1, \dots, d - \alpha_n$ is the eigenvalues of L. Hence, it suffices to find the eigenvalues of A.

For $U = K_n$, the eigenvalues of adjacency matrix A of K_n are $n, -1, \dots, -1$ so the eigenvalues of L are $n, \dots, n, 0$ showing again that K_n has n^{n-2} spanning trees.

Indeed, note that $A + I = J$, the matrix whose entries are all equal to 1. This matrix has rank 1 so $n-1$ of its eigenvalues are 0. As the trace of J (sum of diagonal entries) is n and the trace is also sum of all eigenvalues of J, the remaining eigenvalue must be n. Since $A = J - I$ so eigenvalues of A are eigenvalues of J decreased by 1, and we are done.

6.4 Exercises

Problem 6.4.1. Let $n \geq 2$ be an integer, and let $a_1 \geq \ldots \geq a_n$ be a sequence of positive integers satisfying $a_1 + \ldots + a_n = 2n - 2$. Prove that there exists a tree T on n vertices so that the ordered degree sequence of T is $a_1, \dots, a_n$.

Proof. Induction on n. Since $a_1, \cdots a_n$ are positive integers whose sum is $2n-2$ so $a_{n-1} = a_n = 1$ and $a_1 \geq 2$. By inductive hypothesis, there exists a tree T_{n-1} with degree sequence $(a_1 - 1, a_2, \dots, a_{n-1})$ (may be unordered but that does not matter). We then add a new vertex to T and connect it with a vertex in T_{n-1} whose degree is $a_1 - 1$. $\square$

Problem 6.4.2. Prove that for all $n \geq 3$, the number of t_n of non-isomorphic trees on n vertices is at least $p(n-2)$.

Hint. Use previous problem. □

Problem 6.4.3 (Acyclic function)**.** A function $f : [n] \to [n]$ is called *acyclic* if there are no cycles longer than one under its action on $[n]$. Prove that the number of acyclic functions on $[n]$ is $(n+1)^{n-1}$.

Sketch. Each such function $f : [n] \to [n]$ corresponds to a rooted forests on $[n]$ where the roots are exactly $i \in [n]$ such that $f(i) = i$. In particular, vertex i is adjacent to j if $f(i) = j$. □

Problem 6.4.4 (Parking function)**.** There are n parking spots on $1, \ldots, n$ on a one-way street. Cars $1, \ldots, n$ arrive in this order. Each car i has a favourite parking spot $f(i)$. When a car arrives, it first goes to its favrourite spot. If the spot is free, the car will take it, if not, it goes to the next spot. Again, if that spot is free, the car will take it, if not, the car goes to the next spot. If a car had to leave even the last spot and did not find the space, then its parking attempt has been unsuccessful.

If, at the end of this procedure, all cars have a parking spot, we say that f is a *parking function* on $[n]$. Prove that number of parking functions on $[n]$ is $(n+1)^{n-1}$.

References

bona_walk_to_com [1] Miklos Bona *A Walk Through Combinatorics*

bona_mclenna_white [2] Bóna, M. , Mclennan, A. and White, D. (2000), Permutations with roots. Random Structure Algorithms, 17 (2000) 157-167. doi:10.1002/1098-2418(200009)17:2<157::AID-RSA4>3.0.CO;2-2.