1.
Permutation Statistics

Descents

Notation 1.1. $[n]$
For convenience, fix $n \in \mathbb{N}$ throughout, and we are going to write $[n]$ to denote $\{1, \ldots, n\}$.

Def'n 1.2. **Symmetric Group, Permutation** on $[n]$
The set of bijections from $[n]$ to itself is called the ***symmetric group*** on $[n]$, denoted as S_n. An element $\sigma \in S_n$ is called a ***permutation***.

(1.1) One-line Notation for Permutations

Suppose that $\sigma \in S_n$ is given. To uniquely describe σ, observe that it is enough to write the sequence of values $\sigma(1), \ldots, \sigma(n)$. We often write these values in order without any punctuation to obtain the *one-line notation* for σ:

One-line Notation

$$\sigma(1) \cdots \sigma(n). \tag{1.1}$$

Using the notation in [1.1], the elements of S_3 are:

$$123, 132, 213, 231, 312, 321.$$

Observe that one-line notation gives a bijection between S_n and the collection of lists of $[n]$. Since there are $n!$ lists of $[n]$, it follows that there are

Number of Permutations

$$|S_n| = n! \tag{1.2}$$

permutations on $[n]$.

(1.2) Permutation Statistics

Given a finite set, we are interested in assignming a weight function to the set so that we can wrtie a generating series. In most cases, there is an *obvious* choice of weight function to use. For permutations, however, there are many important – and nontrivial – weight functions (called ***permutation statistics***) to consider.

Def'n 1.3. **Descent, Descent Number** of a Permutation
Let $\sigma \in S_n$. A ***descent*** of σ is $i \in [n-1]$ such that $\sigma(i+1) < \sigma(i)$. We write $\operatorname{des}(\sigma)$ ot denote the number of descents of σ, caleld the descent number of σ.

Def'n 1.4. **Eulerian Polynomial, Eulerian Number**
The ***Eulerian polynomial*** for S_n, denoted as $S_n(x)$, is the generating series for S_n with respecto to des:

$$S_n(x) = \sum\nolimits_{\sigma \in S_n} x^{\operatorname{des}(\sigma)}. \tag{1.3}$$

The coefficients of an Eulerian polynomial are called ***Eulerian number***. For each $k \in [n] \cup \{0\}$, we write $\left\langle {n \atop k} \right\rangle$ to denote

$$\left\langle {n \atop k} \right\rangle = \left[x^k\right] S_n(x). \tag{1.4}$$

(1.3) Eulerian Numbers

Similar to binomial coefficients, there is a nice recurrence relation for Eulerian numbers:

Recurrence Relation for Eulerian Numbers

$$\left\langle {n \atop k} \right\rangle = (n-k)\left\langle {n-1 \atop k-1} \right\rangle + (k+1)\left\langle {n-1 \atop k} \right\rangle. \qquad [1.5]$$

[1.5] follows from observing the fact that we can construct $\sigma \in S_n$ by first defining $\sigma' \in S_{n-1}$, write σ' doen in one-line notation, and insert n at the beginning, end, or in between any two numbers (so n possible *slots* in total). When there are $k-1$ descents of σ', then there are $(n-1)-(k-1) = n-k$ slots to insert n so that $\operatorname{des}(\sigma) = n$. When there are k descents of σ', there are $k+1$ places to insert n so that $\operatorname{des}(\sigma) = n$. Thus [1.5] follows.

Cycles and Left to Right Maxima

Def'n 1.5. **Cycle** of a Permutation

Let $\sigma \in S_n$. A ***cycle*** of σ is a minimal (with respect to $\subseteq$) subset $C \subseteq [n]$ such that $\sigma(C) = C$. We write $\operatorname{cyc}(\sigma)$ to denote the number of cycles of σ.

(1.4) Cycles

Cycles are important because any permutation $\sigma \in S_n$ has a property that the collection of cycles of σ is a partition of $[n]$. That is,

$$\bigcup_{C \subseteq S_n : C \text{ is a cycle of } \sigma} C = [n] \qquad [1.6]$$

is a disjoint union. The generating series for S_n with respect to cyc is of our interest also.

Def'n 1.6. **Signless Stirling Number of the First Kind**

Consider the generating series

$$\Phi_{S_n,\mathrm{cyc}}(x) = \sum\nolimits_{\sigma \in S_n} x^{\operatorname{cyc}(\sigma)}$$

for S_n with respect to cyc. Given any $k \in [n]$, the coefficient

$$c(n,k) = \left[x^k\right] \Phi_{S_n,\mathrm{cyc}}(x)$$

is called a ***signless Stirling number of the first kind***.[a]

[a] We shall say simply *Stirling number* for brevity.

Note that there can be at least 1, at most n cycles of $\sigma \in S_n$. Therefore,

Generating Series for S_n with respect to cyc

$$\Phi_{S_n,\mathrm{cyc}}(x) = \sum\nolimits_{k=1}^{n} c(n,k)\, x^k \qquad [1.7]$$

is the generating series for S_n with respect to cyc. We also have a recurrence relation

Recurrence Relation for Stirling Numbers

$$c(n,k) = c(n-1,k-1) + (n-1)\, c(n-1,k) \qquad [1.8]$$

for Stirling numbers.[1]

[1] The idea of coming up and proving recurrence relations like [1.5] and [1.8] is very general: we consider arbitrary element of S_{n-1}, written in one-line notation, and see what happens when we insert n in the n available slots.

(1.5)
Left to Right Maxima

Another permutation statistic that we take a look is l2r.

Def'n 1.7. **Left to Right Maximum** of a Permutation
Let $\sigma \in S_n$. A ***left to right*** maximum of σ is $i \in [n]$ such that $\sigma(j) < \sigma(i)$ for all $j \in [i-1]$. We write $\mathrm{l2r}(\sigma)$ to dentoe the number of left to right maxima of σ.

An interesting fact about the permutation statistics cyc and l2r is that they have the same generating series. To see this, we are going to look at yet different notation to write dowen a permutation uniquely: by using cycles. First, note that given a cycle $C = \{c_1, \ldots, c_p\}$, we have a list of mappings

$$\begin{gathered} c_1 \mapsto c_2 \\ \vdots \\ c_{p-1} \mapsto c_p \\ c_p \mapsto c_1 \end{gathered},$$

which can be compactly written as

$$c_1 \cdots c_p. \tag{1.9}$$

To distinguish [1.9] from one-line notation, we put paranthesis around:

Cycle Notation

$$(c_1 \cdots c_p). \tag{1.10}$$

But there are p choses for the first element (hence p possible cycle notations for $\sigma|_C$), so to make sure that we have a single cycle notation per cycle, we let $c_1 = \max(C)$. Now we are ready to define the standard cycle notation for permutations.

Def'n 1.8. **Standard Cycle Notation** of a Permutation
Let $\sigma \in S_n$ and let $C_1, \ldots, C_k \subseteq [n]$ be the cycles of σ, where $k = \mathrm{cyc}(\sigma)$, sorted in the increasing order of the maximum element.[a] The ***standard cycle notation*** of σ is

$$\left(m_1 \sigma(m_1) \cdots \sigma^{n_1 - 1}(m_1)\right) \cdots \left(m_k \sigma(m_k) \cdots \sigma^{n_k - 1}(m_k)\right), \tag{1.11}$$

where $n_j = \left|C_j\right|, m_j = \max(C_j)$ for each $j \in [k]$.

[a] That is, given $i, j \in [k]$ with $i < j$, $\max(C_i) < \max(C_j)$.

(1.6)
Cycles and Left to Right Maxima

An interesting feature of the standard cycle notation is that we can remove the paranthesis and still can uniquely determine what the cycles are, thus permutation. For, $m_1 < \cdots < m_k$ and

$$m_j = \max_{p \in \{0, \ldots, n_j - 1\}} \sigma^p(m_j)$$

for all $j \in [k]$, it follows that every left to right maximum of σ is the beginning of a cycle, whose end is right before the next left to right maximum. This means, if we write $\zeta : S_n \to S_n$ to denote the function which takes an element of S_n in one-line notation and spits out its standard cycle notation (without paranthesis), then we have

cyc and l2r

$$\mathrm{cyc}(\sigma) = \mathrm{l2r}(\zeta(\sigma)). \tag{1.12}$$

But recall that $\zeta(\sigma)$ uniquely determines σ. This means ζ is a bijection, so

Generating Series with respect to cyc and l2r

$$\sum_{\sigma\in S_n} \mathrm{cyc}(\sigma) = \sum_{\sigma\in S_n} \mathrm{l2r}(\sigma), \qquad [1.13]$$

the equality that we initially claimed. This in particular means

Coefficients of the Generating Series with respect to l2r

$$\left[x^k\right]\sum_{\sigma\in S_n} \mathrm{l2r}(\sigma) = c(n,k) \qquad [1.14]$$

for all $k \in [n]$.

Major Indices and Inversion Numbers

(1.7) Major Indices

One problem with using des solely is that we lose information about where descents actually occur. Although it is difficult (or even impossible) to express every descents of a permutation by a single statistic, we can still record the sum.

Def'n 1.9. **Major Index** of a Permutation
Let $\sigma \in S_n$. The ***major index*** of σ, denoted as $\mathrm{maj}(\sigma)$, is the sum of the descents of σ.

For instance, $\sigma = 13846725$ has $\mathrm{maj}(\sigma) = 9$. Recall that[2] we found nice recurrence formula for other statistics by using the fact that we can construct an element of S_n by inserting n into any of n slots of $\sigma' \in S_{n-1}$. The same idea applies here: given $\sigma' \in S_{n-1}$ with $k = \mathrm{maj}(\sigma')$, by inserting n in any of n slots, we obtain an element of S_n whose major index is oen of $k, \dots, k+n-1$, where each number occurs exactly once.[3] This gives the following recurrence relation for maj:

Recurrence Relation for maj

$$\sum_{\sigma\in S_n} q^{\mathrm{maj}(\sigma)} = \sum_{i=0}^{n-1} q^i \sum_{\sigma'\in S_{n-1}} q^{\mathrm{maj}(\sigma')}. \qquad [1.15]$$

The polynomial $\sum_{i=0}^{n-1} q^i$ that appears in [1.15] has a particular importance, so we have a name and notation for it.

Def'n 1.10. ***q*-integer**
A polynomial of the form $\sum_{i=0}^{n-1} q^i$, where $n \in \mathbb{N}$, is called a ***q-integer***. We write $[n]_q = \sum_{i=0}^{n-1} q^i$.

Using the notation $[n]_q$, [1.15] becomes

Generating Series with respect to maj

$$\sum_{\sigma\in S_n} q^{\mathrm{maj}(\sigma)} = [n]_q \cdots [1]_q. \qquad [1.16]$$

Note that the expression at the RHS of [1.16] looks like a factorial: we call it a ***q-factorial***, denoted as $[n]_q!$.

[2] See [1.5], [1.8], especially the footnote after [1.8].

[3] An easy argument for this line is a proof by construction. Begin with placing n at the descent positions, from right to left, and then at the ascent positions, from left to right.

Def'n 1.11. **Mahonian** Statistic

A permutation statistic is called ***Mahonian*** if the generating series for S_n with respect to the statistic is a q-factorial.

(1.8) Inversion Numbers

Clearly maj is Mahonian. But arguably the most important Mahonian statistic is the inversion number.

Def'n 1.12. **Inversion, Inversion Number** of a Permutation

Let $\sigma \in S_n$. An ***inversion*** of σ is a pair $(i, j) \in [n]^2$ with $i < j$ such that $w(i) > w(j)$. The ***inversion number*** of σ, denoted as $\operatorname{inv}(\sigma)$, is the number of inversions σ has.

In fact, given any $\sigma' \in S_{n-1}$, we can produce $\sigma \in S_n$ with $\operatorname{inv}(\sigma) = k + i$ by placing n at the $(n-i)$th slot from the beginning of the one-line notation of σ'. This means inv has the precisely same recurrence relation as maj:

Recurrence Relation for inv

$$\sum\nolimits_{\sigma \in S_n} q^{\operatorname{inv}(\sigma)} = [n]_q \sum\nolimits_{\sigma' \in S_{n-1}} q^{\operatorname{inv}(\sigma')}. \tag{1.17}$$

This means inv is also Mahonian, as said before:

Generating Series with respect to inv

$$\sum\nolimits_{\sigma \in S_n} q^{\operatorname{inv}(\sigma)} = [n]_q!. \tag{1.18}$$

Since $\Phi_{S_n,\mathrm{maj}}(q) = \Phi_{S_n,\mathrm{inv}}(q)$, it is natural to expect that there is a bijection $\varphi : S_n \to S_n$ such that

$$\operatorname{maj}(\sigma) = \operatorname{inv}(\varphi(\sigma))$$

for all $\sigma \in S_n$, similar to the case of cyc and l2r. But finding such bijection is much difficult for this case.

Algebraic Aspects of the Symmetric Group

(1.9) Note that the symmetric group S_n is a *group* under the following operation: given any $\sigma, \tau \in S_n$, write $\sigma\tau$ (the ***multiplication*** of σ, τ) to denote

$$\sigma\tau = \sigma \circ \tau,$$

since we can compose any elements of S_n, obtaining an element of S_n again. Moreover, we have the following.

(a) *identity*: There is an ***identity*** element of S_n, denoted as $\mathrm{id} \in S_n$, such that

$$\mathrm{id}\,\sigma = \sigma\,\mathrm{id} = \sigma$$

for all $\sigma \in S_n$, namely $\mathrm{id} = 1 \cdots n$ (in one-line notation).

(b) *inverse*: Given any $\sigma \in S_n$, there exists $\tau \in S_n$ such that

$$\sigma\tau = \tau\sigma = \mathrm{id}. \tag{1.19}$$

Note that clearly $\tau = \sigma^{-1}$ (inverse as a function) works, and by [1.19] this τ is unique.

(c) *associativity*: Given $\sigma, \tau, \rho \in S_n$, we have

$$(\sigma\tau)\rho = \sigma(\tau\rho) = \sigma\tau\rho.$$

An important type of permutation is a simple transposition.

Simple Transposition

Def'n 1.13. A permutation $\sigma \in S_n$ is called a ***simple transposition*** if there exists $i \in [n-1]$ such that

$$\sigma(j) = \begin{cases} j+1 & \text{if } j = i \\ j-1 & \text{if } j = i+1 \\ j & \text{if } j \notin \{i, i+1\} \end{cases}. \qquad [1.20]$$

The unique simple transposition in S_n that satisfies [1.20] is denoted as s_i.

Multiplying by a simple transposition is easy in terms of one-line notation: given $\sigma \in S_n$,

(a) σs_i is obtained by swapping the letters in *positions* $i, i+1$ of the one-line notation of σ; and

(b) σs_i is obtained by swapping the letters in *values* $i, i+1$ of the one-line notation of σ.

Note that, given any $\sigma \in S_n$, we can write σ as a product of simple transpositions. We also have the following formula:

Inversion Number after Multiplying by a Simple Transposition

$$\operatorname{inv}(\sigma s_i) = \begin{cases} \operatorname{inv}(\sigma) + 1 & \text{if } \sigma(i) < \sigma(i+1) \\ \operatorname{inv}(\sigma) - 1 & \text{if } \sigma(i) > \sigma(i+1) \end{cases}. \qquad [1.21]$$

This page intentionally left blank.

2.
Permutations and Graphs

Reduced Words

(2.1) Reduced Words

Suppose that a permutation $w \in S_n$ is given. Then recall that we can write w as a product of simple transpositions

$$w = s_{i_1} \cdots s_{i_k}, \qquad [2.1]$$

and that $k \geq \operatorname{inv}(w)$. In fact, we know that we can use exactly $\operatorname{inv}(w)$ simple transpositions.

Def'n 2.1. **Reduced Decomposition, Reduced Word** for a Permuataion
An expression of the form [2.1] with $k = \operatorname{inv}(w)$ is called a ***reduced decomposition*** for w, and the sequence $(i_j)_{j=1}^{k}$ is called a *reduced word* for w. We write $R(w)$ to denote the set of the reduced words of w.

(EX 2.2) Suppose that $w = 31542 \in S_5$, then $42134, 23413$ are reduced words for w.

(2.3) Generating Reduced Words

Given $w \in S_n$, we desire to calculate $|R(w)|$. But before doing so, we are going to discuss about systematically writing down every element of $R(w)$, by using graphs. A brute-force way of achieving this is to look at every string in $\{1, \ldots, n-1\}^n$ and see check which one is a reduced word for w. Of course this is very inefficient, so we are going to find a better algorithm. We start by finding a reduced word, from where we are going to find everything else. But we already know how to get a reduced word, so it suffices to find relations between reduced words.

(a) Given $i, j \in [n-1]$, if $|i-j| \geq 2$, then $s_i s_j = s_j s_i$. Such *swap* is called a ***commutation***.

(b) Given $i \in [n]$, observe that $s_i s_{i+1} s_i = s_{i+1} s_i s_{i+1}$. Such *swap* is called a ***braid move***.

It turns out by using commutations and braid moves only, we can generate every reduced word of a permutation from a reduced word.

Theorem 2.1. Generating Reduced Words

Let $w \in S_n$ and let $r, s \in R(w)$. Then s can be obtained from r by a sequence of commutations and braid moves.

(2.4) Graph of Reduced Words

Fix $w \in S_n$ and consider the $G(w)$ whose vertex set is $R(w)$ and vertices $v, u \in R(w)$ are adjacent if and only if one can be turned into another by a commutation or braid move.

Theorem 2.2. Connectedness of $G(w)$

Consider the setting of (2.4). $G(w)$ is connected.

Lemma 2.2.1. Exchange Lemma

If $i_1 \cdots i_l, j_1 \cdots j_l \in R(w)$, then

$$j_1 i_1 \cdots i_{k-1} i_{k+1} \cdots i_l \in R(w)$$

for some $k \in [l]$.

Proof. Observe that $i_l \cdots i_1, j_l \cdots j_1 \in R\left(w^{-1}\right)$. Then from assignments, we know the following.

(a) $(j_1, j_1 + 1)$ is an inversion of w^{-1}.

(b) $(j_1, j_1+1) = s_{i_1} \cdots s_{i_{k-1}}(i_k, i_k+1)$ for some $k \in [l]$.[1] This means *flipping* j_1, j_1+1 *before* $s_{i_1} \cdots s_{i_{k-1}}$ is equivalent to first applying $s_{i_1} \cdots s_{i+k-1}$ and then *flipping* i_k, i_k+1. In other words,

$$s_{j_1} s_{i_1} \cdots s_{i_{k-1}} = s_{i_1} \cdots s_{i_k}.$$

Then observe that

$$s_{j_1} s_{i_1} \cdots s_{i_{k-1}} s_{i_{k+1}} \cdots s_{i_l} = s_{i_1} \cdots s_{i_l},$$

as desired. ∎

Proof of Theorem 2.2. We proceed inductively.

- When $w \in S_n$ is such that $\operatorname{inv}(w) = 0$ or $\operatorname{inv}(w) = 1$, then $G(w)$ is clearly connected. ◁
- Suppose that the result holds for permutations whose inversion number is $l-1$, where $l \in \mathbb{N}$. Let $i_1 \cdots i_l, j_1 \cdots j_l \in R(w)$ and denote $i = i_1 \cdots i_l, j = j_1 \cdots j_l$. Then by the exchange lemma, $j_1 i_1 \cdots i_{k-1} i_{k+1} \cdots i_l \in R(w)$ for some $k \in [l]$, and denote $e = j_1 i_1 \cdots i_{k-1} i_{k+1} \cdots i_l$. The idea is to find an i,e-path and then find an e,j-path. We have two cases.

  (a) Suppose that $k < l$. Then by deleting i_l, we obtain

  $$\underbrace{i_1 \cdots i_{l-1}}_{=i'}, \underbrace{j_1 i_1 \cdots i_{k-1} i_{k+1} \cdots i_{l-1}}_{=e'},$$

  which are reduced words for ws_{i_l}. So by the inductive hypothesis, there exists a i',e'-path in $G(ws_{i_l})$. But note that every edge in any i',e'-path consists of commutations and braid moves between the $l-1$ indices of i',e', which are the first $l-1$ indices of i,e. This means the same commutation and braid edges can be used to produce an i,e-path. Obtaining an e,j-path can be done in a similar manner, by first deleting j_1 from the front and then using the inductive hypothesis.

  (b) Suppose that $k = l$. In this case, we are going to find another vertex, which would allow us to use the idea used in case (a). Note that

  $$e = j_1 i_1 \cdots i_{l-1}.$$

  We have two subcases.

  (i) If $|j_1 - i_1| < 2$, then $e, \underbrace{i_1 j_1 i_2 \cdots i_{l-1}}_{=f}$ are neighbors, and we can use the same idea as (a) to find a i,f-path and e,j-path, so we are done.

  (ii) If $|j_1 - i_1| = 1$, we apply the exchange lemma again, on i,e. That is, there exists $h \in [l-1]$[2] such that

  $$\underbrace{i_1 j_1 i_1 \cdots i_{h-1} i_{h+1} \cdots i_{l-1}}_{=f}$$

  is a reduced word for w. Since i,f both start with i_1, by using the same idea used in (a), we obtain an i,f-path. But note that f is a neighbor of

  $$\underbrace{j_1 i_1 j_1 i_2 \cdots i_{h-1} i_{h+1} \cdots i_{l-1}}_{=g}$$

  by a braid edge (since $|i_1 - j_1| = 1$ by assumption), and we can find a g,e-path since they both start by j_1. This concludes the proof. ∎

---

[1] $s_k(a,b)$ means the ordered pair obtained from (a,b) by replacing any instance of k with $k+1$ and any instance of $k+1$ with k.

[2] We better not remove the first character of e, since then we have $i_1 i_1 \cdots$, which clearly is not a reduced word.

Platonic Solids

(2.5) Platonic Solids

Recall that we have the following five *Platonic solids*: *tetrahedron, cube, octahedron, dodecahedron, icosahedron*. We are going to associate each Platonic solid to a group that *looks like a symmetric group*. The idea of *duality* also plays an important role.

- *tetrahedron*: Note that tetrahedron is the planar dual of itself. Tetrahedron corresponds to S_4.
- *cube, octahedron*: Observe that cube and octahedron are planar duals. As a result, they both correspond to a *hyperoctahedral group*, denoted as B_n, the group of *signed permutations*.
- *dodecahedron, icosahedron*: Dodecahedron and icosahedron are planar duals, and they both correspond to H_3.

We are going to discuss signed permutations mainly.

Def'n 2.2. **Signed Permutation** on $[n]$

Let $n \in \mathbb{N}$. A ***signed permutation*** on $[n]$ is a permutation w on $\{-n, \ldots, n\}$ with an additional property that

$$w(i) = -w(-i)$$

for all $i \in [n]$. We write B_n to denote the set of signed permutations on $[n]$.

By definition, note that it is enough to just define $w(1), \ldots, w(n)$ to define w for all $\{-n, \ldots, n\}$. Therefore, we usually think a signed permutation on $[n]$ to a permutation on $[n]$ with an additional information (i.e. sign) on each number. Correspondingly, we have the following way of writing ***one-line notation*** for a signed permutation. Given a signed permutation w on $[n]$, we first write the one-line notation which disregards the signs, and for every negative $w(i)$, we put a bar above. So, for instance,

$$21\overline{3}4\overline{5}\overline{9}78\overline{6}$$

is a one-line notation for a signed permutation. Now consider the following question.

Given $n \in \mathbb{N}$, how many signed permutations on $[n]$ are there? [2.2]

Note that the answer is $2^n n!$, since we have 2^n ways of choosing which $w(i)$ would be negative and $n!$ ways of choosing *unsigned* permutations. That is,

$$|B_n| = 2^n n!.$$

3.
Matroids

Matroids

(3.1) Motivation

Matroid theory serves as common generalization of graph theory and linear algebra.

Finite Matroid on a Set

Def'n 3.1. Let E be a finite set. We say a collection M of subsets of E is a ***matroid*** on E if

- $M \neq \varnothing$;
- if $X \in M, Y \subseteq X$, then $Y \in M$; and
- if $X, Y \in M$ are maximal and $e \in X \setminus Y$, then there exists $f \in Y \setminus X$ such that $(X \setminus \{e\}) \cup \{f\}$ is maximal.

We shall discuss finite matroids only. In other words, when we say *matroid*, we mean *finite matroid*.

Independent, Dependent Subset

Def'n 3.2. Let E be a finite set and let M be a matroid on E.

(a) $X \subseteq E$ is called ***independent*** if $X \in E$. Otherwise, we say X is ***dependent***.

(b) Maximal $X \in M$ is called a ***basis***, and a minimal $Y \in \mathcal{P}(E) \setminus M$ is called a ***circuit***.

We define the ***rank function*** on E, rank : $\mathcal{P}(E) \to \mathbb{N} \cup \{0\}$, by

$$\operatorname{rank}(X) = \max\{|Y| : Y \subseteq X, Y \in M\}$$

for all $X \subseteq E$. We call $\operatorname{rank}(X)$ the ***rank*** of X.

Loop, Parallel Elements of a Matroid

Def'n 3.3. Let E be a finite set and let M be a matroid on E.

(a) A ***loop*** of M is a singleton $\{e\} \subseteq E$ such that $\{e\} \notin M$.

(b) ***Parallel elements*** are elements $e, f \in E$ such that e, f are not loops but $\{e, f\} \notin M$.

Vector Matroid

Def'n 3.4. Let V be a vector space over a field $\mathbb{K}$ and let $E \subseteq V$ be finite. Then

$$M = \{X \subseteq E : X \text{ is linearly independent}\} \tag{3.1}$$

is called the ***vector*** matroid over E.

(3.2) Vector Matroids

We have to verify that [3.1] actually defines a matroid.

(a) $\varnothing \in M$, so M is nonempty.

(b) If $X \subseteq E$ is linearly independent, then any $Y \subseteq X$ is linearly independent also, so any $Y \subseteq X$ is an element of E.

(c) Note that maximally linearly independent subsets of E are the bases of $\operatorname{span}(E)$. So if $X, Y \in M$ are maximal, then $|X| = |Y| = \underbrace{\dim(\operatorname{span}(E))}_{=d}$. So if $e \in X \setminus Y$, then $\operatorname{span}(X \setminus \{e\})$ is $(d-1)$-dimensional, which means there exists $f \in Y$ that is not in $\operatorname{span}(X \setminus \{e\})$. But note that this f

is not in X, since otherwise $f \in \operatorname{span}(X \setminus \{e\})$. Thus we found $e \in X \setminus Y, f \in Y \setminus X$ such that $(X \setminus \{e\}) \cup \{f\}$ is a basis for E.

Def'n 3.5. **Graphical** Matroid

Let G be a *multi*-graph[a] with edge set E. Let M be the collection of subsets of E such that given any $X \in M$, deleting all edges not in X gives a forest. In other words, if $X \in M$, then

$$(V(G), X)$$

is a forest. We call M the ***graphical*** matroid over E.

[a] In other words, we allow multiple edges incident to the same vertices and loops.

(3.3) Graphical Matroids

We show that graphical matriods are matriods.

(a) $\varnothing \in M$, so M is nonempty.

(b) If $X \in M$, then given any $Y \in X$, $Y \in M$, since we already deleted enough edges to form an acyclic graph with X, and we are deleting even more with Y.

(c) Note that maximal elements of M are spanning forests. Suppose that $X, Y \in M$ are maximal. Then given any $e \in X \setminus Y$, note that we are *spliting* a component of X into 2 components by removing e from X. But since $e \notin Y$, there exists $f \in Y$ joining these pieces, so $X - e + f$ is again a spanning forest. Thus $X - e + f$ is a matroid.

(3.4) Analogy between Matroids, Vector Spaces, and Graphs

We have the following table that shows analogy between matroids, vector spaces, and graphs.

Analogy between Matroids, Vector Spaces, and Graphs

matroids	vector spaces	graphs
independent sets	independent sets	subforests
dependent sets	dependent sets	subgraphs with a cycle
bases	bases of $\operatorname{span}(E)$, where E is a finite subset of the vector space	spanning forests
rank	$\dim(\operatorname{span}(E))$	maximum number of edges in a subforest
circuits	minimal dependent subsets	cycles
loops	$\{\mathbf{0}\}$	loops
parallel elements	parallel vectors	multiple edges

Def'n 3.6. **Free** Matroid

Let E be a finite set. The ***free*** matroid M on E is the collection of subsets of E: $M = \mathcal{P}(E)$.

(3.5) Free Matroid

Note that any free matroid is a matroid.

(a) Clearly M is nonempty.

(b) Clearly if $X \subseteq E$ and $Y \subseteq X$, then $Y \subseteq E$.

(c) E is the only maximal independent set, so the last condition is vacuously true.

In particular, E is the basis for M. It turns out that a free matroid is a vector matroid and a graphical matroid.

Greedy Algorithms for Matroids

(3.6)
Prim's Algorithm for MST

Recall that *Prim's algorithm* successfully finds an MST for any connected weighted graph.

(a) Start with any vertex and empty edge set.

(b) At any iterative step, consider the cut induced by the current set of vertices. Choose the edge with the least weight, include the edge in the edge set, and vertex which is incident to the edge and not currently in the vertex set to the vertex set.

We claim that there is an analogous algorithm works in the matroid setting. Before proving this, we introduce some notions.

Def'n 3.7. **Simplicial Complex** on a Set

Let E be a finite set. A ***simplicial complex*** I on E is a collection of subsets of E such that

(a) $\varnothing \in I$; and

(b) if $X \in I$, then $Y \in I$ for any $Y \subseteq X$.

(3.7)
Simplicial Complexes

Note that a matroid is a simplicial complex with an additional notion of basis.

Def'n 3.8. **Isomorphism** of Simplicial Complexes

Let E,F be finite sets and let I be a simplicial complex on E, J be a simplical complex on F. If bijective $\varphi : E \to F$ satisfies

$$X \in I \iff \varphi(X) \in J,$$

then we say φ is an isomorphism of I,J.

There are other ways of specifying a simplicial complex. Suppose that I is a simplicial complex on a finite set E. Then we can uniquely determine what I is if we know any one of the following:

(a) I;

(b) maximal elements of I;

(c) $\mathcal{P}(E) \setminus I$; and

(d) minimal elements of $\mathcal{P}(E) \setminus I$.

Theorem 3.1.
Characterizations of Matroids

Let E be a finite set and let I be a simplicial complex of E. Write

- *B to denote the collection of maximal elements of I;*
- *F to denote $\mathcal{P}(E) \setminus I$; and*
- *C to denote the collection of minimal elements of F.*

Then the following are equivalent.

(a) I is a matroid (i.e. if $X,Y \in B$ and $e \in X \setminus Y$, then there is $f \in Y \setminus X$ such that $X \setminus \{e\} \cup \{f\} \in B$).

(b) For every $X,Y \in B$ and $e \in X \setminus Y$ there exists $f \in Y \setminus X$ such that $Y \cup \{e\} \setminus \{f\} \in B$.

(c) For any $E' \subseteq E$, if $X,Y \subseteq E'$ are maximal in $\mathcal{P}(E') \cap I$, then $|X| = |Y|$.

(d) If $X,Y \in I$ and $|X| < |Y|$, then there exists $y \in Y \setminus X$ such that $X \cup \{y\} \in I$.

(e) For every nonnegative weight function on E, there is a greedy algorithm which finds a maximum weight independent set.