Chapter 1 - Basic Principles

- Cartesian product: $A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$
  - This generalizes to A^k
  - A^0 is the set containing the empty string $\{\varepsilon\}$
- Size of cartesian product: $|A \times B| = |A| \times |B|$
- Size of union: $|A \cup B| = |A| + |B| - |A \cap B|$
  - Disjoint union: $|A \cup B| = |A| + |B|$
  - Inclusion exclusion (multiple unions): Subtract intersections of an even number of sets, add intersections of intersections of an odd number of sets.
  - $|A_1 \cup \cdots \cup A_m| = \left|\bigcup_{i=1}^{m} A_i\right| = \sum_{\emptyset \neq S \subseteq \{1,2,\ldots m\}} (-1)^{|S|-1} |A_S|$
- There are $n!$ lists of an n-element set
- There are 2^n subsets of an n-element set
- There are $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ k-element subsets of n elements

> **Binomial theorem**
>
> $$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = \sum_{x=0}^{n} \binom{n}{x} = 2^n$$
>
> Motivation: this is the number of ways to count 0-element subsets + 1 element subsets + ... + n-element subsets, so it is the same as counting all the subsets

- $\binom{n}{k} = \binom{n}{n-k}$, since we choose the n items *not to pick*
- Pascal's Identity: $\binom{n}{k} = \binom{n}{k-1} + \binom{n-1}{k-1}$

> **Multisets**
>
> For any $n \geq 0$ and $t \geq 1$, there are $\binom{n+t-1}{t-1}$ n-element *multisets* with t types

- Bijections (we assume $f : \mathcal{A} \to \mathcal{B}$, $a, a' \in \mathcal{A}$ and $b \in \mathcal{B}$)
  - Surjective: for every b there exists an a such that $f(a) = b$
  - Injective: If $f(a) = f(a')$, then we must have $a = a'$
  - Bijective: f is both injective and surjective (then $\mathcal{A} \leftrightarrow \mathcal{B}$)
  - Mutually inverse bijection: $f^{-1} : \mathcal{B} \to \mathcal{A}$
- Bijective proof: show two sets have the same size by establishing a bijection between them
  - Provide the function that maps one set to the other and its inverse (define f and f^{-1})

    - (Possibly) prove that $f(f^{-1}(x)) = f^{-1}(f(x)) = x$

Question strategies

- Figure out exactly what each side of the equation counts (choose -> selecting subsets, 2^x -> all possible subsets, multiplication -> and then, addition -> or, etc.), then think of a way to make one side count the set of the other (**combinatorial proof**)
    - For one thing *and* another thing (possibly more): use a pair or set (ex. (n, A) where $n \in \mathbb{N}$ and A is in some set)
    - Accounting for order overlaps: declare that a sequence is sorted in order
- **Bijective proofs**
    - Write out everything and try to find a pattern that turns items in one set to another
    - Then write function that links the sets
- **Indicator vector**: bijection between subsets of a set of size n and the binary strings of length n: each digit in the string is 1 if the corresponding element is in the subset and 0 otherwise
    - Indicator vectors that sum to $k \leftrightarrow$ subsets of size k
- Establish that a union is disjoint before counting or compensate for the overlap

Chapter 2 - Generating Series

Useful Series

Geometric Series

$$\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n = 1 + x + x^2 + x^3 + \dots$$

Binomial Series

$$(1+x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k$$

- Generating series for the number of subsets of a set of size n

Negative binomial series

$$\frac{1}{(1+x)^t} = \sum_{n=0}^{\infty} \binom{n+t-1}{t-1} x^n$$

- I.e. the binomial series with a negative integer exponent

- Generating series for the number of multisets with $t \geq 1$ types of elements (regardless of size)

Generating Series

- **Weight function**: Function $\omega : \mathcal{A} \to \mathbb{N}$ that encodes the "weight" of each element in the set $\mathcal{A}$ as a natural number
  - There cannot be infinite items of a given weight in a set (i.e. $\mathcal{A}$ is countable infinite)
- **Generating series** encode an ordered sequences of numbers as the coefficients of a power series
  - Ex. $G(x) = g_0 + g_1 x + g_2 x^2 + \dots = \sum_{n=0}^{\infty} g_n x^n$
- **Generating series of a set**: encodes the number of elements of a given weight (with respect to ω) in the set: $A(x) = \phi_{\mathcal{A}}^{\omega}(x) = \sum_{\alpha \in \mathcal{A}} x^{\omega(\alpha)}$
  - I.e. $\phi_{\mathcal{A}}(x) = a_0 + a_1 x + a_2 x^2 + \dots = \sum_{n=0}^{\infty} a_n x^n$ where $a_n = |\mathcal{A}_n|$

- We also say $[x^k]\mathcal{A}(x) = a_k$

Generating Series Lemmas

Sum Lemma

Let A and B be disjoint sets with weight function ω. Then we have $\phi_{A\cup B}(x) = \phi_A(x) + \phi_B(x)$

- I.e. We can add generating series together to get the generating series of the disjoint union of the sets in question
- This works on any number of (possibly infinitely many) mutually disjoint sets (Infinite Sum Lemma)

Product Lemma

Let A and B be sets with the weight functions ω and υ repectively. Define the weight function $\eta : A \times B \to \mathbb{N}$ as $\eta(a,b) = \omega(a) + \upsilon(b)$ for all $(a,b) \in A \times B$. Then

- η is the weight function for $A \times B$
- $\phi^{\eta}_{A\times B}(x) = \phi^{\omega}_A(x) \times \phi^{\upsilon}_B(x)$

- I.e. we can multiply generating series together
- This also shows that a series can be raised to a power: $\phi_{A^k}(x) = (\phi_A(x))^k$
- Multiplication of generating series: $\left(\sum_{n=0}^{\infty} c_n x^n\right)\left(\sum_{n=0}^{\infty} d_n x^n\right) = \sum_{n=0}^{\infty}\sum_{k=0}^{n} c_k d_{n-k} x^n$
- This also works with the cartesian product of any number of sets

String Lemma

Let A be a set with a weight function $\omega : A \to \mathbb{N}$ where no element has weight 0. Then $\phi_{A^*}(x) = \dfrac{1}{1-\phi_A(x)}$ where $A^* = \bigcup_{k=0}^{\infty} A^k$ (set of all cartiesian products of A)

- The weight function ω^* for A^* is $\omega^* = \omega(a_1) + \omega(a_2) + \ldots$ (i.e. the sum of the weights of the elements in each tuple)

Compositions

- **Composition**: finite sequence of strictly positive integers $\gamma = (c_1, c_2, \ldots, c_k)$
  - Here, $k \in \mathbb{N}$ and each $c_i \geq 1 \in \mathbb{Z}$
  - The size (weight) $|\gamma|$ of a composition is the sum of its parts

- Set of all compositions: $\mathcal{C} = \bigcup_{k=0}^{\infty} \{1, 2, \dots\}^k = \{1, 2, \dots\}^* = \bigcup_{k=0}^{\infty} \mathbb{N}_+^k$
- Generating series with respect to size: $\phi_{\mathcal{C}}(x) = 1 + \frac{x}{1-2x}$ (found from string lemma)
- For $n \in \mathbb{N}$, the number of compositions of size n ($[x^n]\phi_{\mathcal{C}}(x)$) is 1 for $n = 0$ and 2^{n-1} otherwise
  - There are $\binom{n-1}{k-1}$ compositions of size n with length k

> **Proposition 2.23**
> There is a bijection betwen $\mathcal{C} \setminus \{\varepsilon\}$ the set of paris (n, A) where $n \in \mathbb{N}$ and $A \subseteq \{1, 2, \dots n\}$

- Bijection (rough): Sort A. The gaps between each element in A correspond to the elements in the composition. The length of the corresponding composition is $|A| + 1$

Question Strategies

- Find the generating series of the set of compositions with simple constraints
  - Determine the generating series of a single part and/or set of allowed parts (call it $\phi(x)$)
  - Describe the set of possible generating series, likely using an infinite union to account for all sizes
  - By product lemma, the generating series for the set of compositions of length k is $(\phi(x))^k$
    - Constraints on length -> build into exponent (ex. odd length -> $\phi(x)^{2j+1}$)
  - By string lemma, whole generating series is $\sum_{k=0}^{\infty} (\phi(x))^k = \frac{1}{1-\phi(x)}$ (solve using algebra)
- Multiple types of parts at different spots (i.e. first item is even, each item is equal to the parity of its index, etc.)
  - Figure out the generating series for each part
  - Figure out how to write the set of all compositions using a cartesian product of the sets of possible items at each index, since the whole composition is just the cartesian product of these. Cases may be required (ex. odd vs. even length)
    - Cases: take the union of the sets, which becomes adding the power series together
  - Use the sum, product, and possibly string lemmas to translate the set definition into a generating series based on the generating series for each part
- Subset with restriction (i.e. how many subsets of $\mathbb{N}$ have a given property)
  - Use proposition 2.23: figure out which property the corresponding composition must have and count that
- Finding $[x^n]$ of nested series: figure out the all the possible combinations of indices of the two series that add to n and add their coefficients

Chapter 3 - Binary Strings

- **Binary string**: finite sequence $\sigma = b_1 b_2 \dots b_n$ where each bit b_i is either 0 or 1
  - So, a string of length n is a member of the set $\{0,1\}^n$, and the set of all binary strings is $\{0,1\}^* = \bigcup_{n=0}^{\infty} \{0,1\}^n$
- There are 2^n binary strings of length n, so the generating series is $\frac{1}{1-2x}$

Regular Expressions and Rational Languages

Regular Expression

A **regular expression** is one of the following:

- ε, 0, or 1
- "Union": If R and S are regular expressions, then R $\smile$ S
- Concatenation: If R and S are regular expressions, then RS
  - We can also have R^k, i.e. R^2 $=$ RR, etc
  - Concatenation product: like cartesian product, except that a string may be produced more than once, so we may have $|\mathcal{RS}| \leq |\mathcal{R} \times \mathcal{S}|$
- If R is a regular expression, then so is R*

- Each regular expression R will *produce* a subset of the set of all binary strings $\mathcal{R} \subseteq \{0,1\}^*$ called a **rational language**
- A regular expression will *lead* to a rational function $R(x)$
  - Sometimes, $R(x)$ is the generating series for $\mathcal{R}$ with respect to length
- Production of rational languages from regular expressions $\mathcal{R} \subseteq \{0,1\}^*$
  - ε produces $\{\varepsilon\}$, 0 produces $\{0\}$ and 1 produces $\{1\}$
  - R $\smile$ S produces $\mathcal{R} \cup \mathcal{S}$
    - This union is not necessarily disjoint
  - RS produces $\mathcal{RS}$ (concatenation product (for longer strings))
  - R* produces $\mathcal{R}^* = \bigcup_{k=0}^{\infty} \mathcal{R}^k$, where $\mathcal{R}^k$ is the concatenation product of k copies of $\mathcal{R}$
- Ex. $(01)^*$ produces $\{\varepsilon, 01, 0101, 010101, \dots\}$

Unambiguous Expressions

- A regular expression R is **unambiguous** iff every string in $\mathcal{R}$ is produced exactly once by R

- ε, 0, and 1 are unambiguous
- R $\smile$ S is unambiguous $\iff$ $\mathcal{R} \cup \mathcal{S}$ is a disjoint union
- RS is unambiguous $\iff$ $|\mathcal{RS}| = |\mathcal{R} \times \mathcal{S}|$ (i.e. no strings are produced more than once)
- R* is unambiguous $\iff$ all R^k are unambiguous and $\bigcup_{k=0}^{\infty} \mathcal{R}^k$ is disjoint

Getting Generating Series

Getting Generating Series from Regular Expressions

Let regular expressions R and S have rational functions $R(x)$ and $S(x)$

- ε leads to 1
- 0 and 1 both lead to x
- R $\smile$ S leads to $R(x) + S(x)$
- RS leads to $R(x)S(x)$
- R* leads to $\dfrac{1}{1 - R(x)}$

- Expressions must be unambiguous for this to work

Decompositions

- **Block**: maximal nonempty subsequence of consecutive equal bits
- $0^*(1^*10^*0)^*1^*$ and $1^*(0^*01^*1)^*0^*$ build up each string in $\{0, 1\}^n$ block by block (both are unambiguous)
- **Prefix decomposition**: rational expression A*B where the string is formed of segments created by A, with a possible terminal segment B
  - Usually unambiguous $\iff$ only one way for string to begin with a segment of A $\wedge$ string is produced by B if does not begin with a segment of A

Recursive Decompositions

- Happens when a regular expression is defined in terms of itself (ex. S $= \varepsilon \smile (0 \smile 1)$S)
  - So, $S(x)$ is defined in terms of $S(x)$ (ex. $S(x) = 1 + (x + x)S(x)$)
  - We can solve using algebra to find a non-recursive definition

Excluded substrings

- σ **contains** κ $\iff$ there exist α, β such that $\sigma = \alpha\kappa\beta$
  - Otherwise, σ *avoids* κ

Excluded Substrings

Let κ be a binary string of length n, and let A_κ be the set of binary strings that avoid κ. Let $\mathcal{C}$ be the set of nonempty suffixes γ of κ such that $\kappa\gamma = \eta\kappa$ for some nonempty prefix η of κ. Let $C(x)$ = $\sum_{\gamma \in \mathcal{C}} x^{\ell(\gamma)}$. Then $A_\kappa(x) = \dfrac{1 + C(x)}{(1-2x)(1+C(x)) + x^n}$

- Essentially, γ are all the ways that κ can overlap with itself (i.e. last n characters are the same as the first n characters)
  - We can use a table to find γ

Question Strategies

- Add a ε somewhere in a regular expression to account for the fact that it may be the empty string if performing a block decomposition

Chapter 4 - Recurrence Relations

Fibonacci Numbers

- Definition: $f_0 = 1, f_1 = 1, f_n = f_{n-1} + f_{n-2}$ for $n \geq 2$
- Solving generating series: $F(x) = f_0 + f_1 x + \sum_{n=2}^{\infty} f_n x^n = 1 + x + \sum_{n=2}^{\infty}(f_{n-1} + f_{n-2})x^n =$ $1 + x + \sum_{n=2}^{\infty} f_{n-1}x^n + \sum_{n=2}^{\infty} f_{n-2}x^n = 1 + x + x(F(x) - f_0) + x^2 F(x) = 1 + xF(x) + x^2 F(x) \implies$ $F(x)\frac{1}{1 - x - x^2}$
- We can use inverse roots and some algebra to find that $f_n \approx \frac{5+\sqrt{5}}{10}\left(\frac{1+\sqrt{5}}{2}\right)^n$ (f_n is the closest integer to this expression)

Homogenous Linear Recurrence Relations

Homogenous Linear Recurrence Relation

Let $g = (g_0, g_1, \ldots)$ be an infinite sequence of complex numbers and let $a_1, a_2 \ldots a_d$ be in $\mathbb{C}$. Let $N \in \mathbb{Z} \geq d$. Then, g satisfies a HLRR if $g_n + a_1 g_{n-1} + a_2 g_{n-2} + \cdots + a_d g_{n-d} = 0$, i.e. $g_n = -(a_1 g_{n-1} + \cdots + a_d g_{n-d})$

Generating series for relations

- For the HLRR $g_n + a_1 g_{n-1} + a_2 g_{n-2} + \cdots + a_d g_{n-d} = 0$, we have $\sum_{k=n}^{\infty}(g_n + a_1 g_{n-1} + a_2 g_{n-2} + \cdots + a_d g_{n-d})x^k = 0$
  - Split this into d different summations
  - Pull the coefficients $a_1 \ldots a_d$ in front of the series
  - Write each summation in terms of the whole power series $G(x)$ (recursive)
  - Derive $G(x)$ using algebra

Theorem 4.8

Let $g = (g_0, g_1, g_2, \ldots)$ be a sequence of complex numbers with generating series $G(x)$. Then

1. g satisfies the HLRR $g_n + a_1 g_{n-1} + a_2 g_{n-2} + \cdots + a_d g_{n-d} = 0$ for all $n \geq N$ with initial conditions $g_0, \ldots, g_{N-1}$
2. The series $G(x) = \frac{P(x)}{Q(x)}$ is a quotient of two polynomials

- Denominator: $Q(x) = 1 + a_1x + a_2x^2 + \cdots + a_dx^d$
- Numerator: $P(x) = b_0 + b_1x + b_2x^2 + \cdots + b_{N-1}x^{N-1}$ where $b_k = g_k + a_1g_{k-1} + \cdots + a_dg_{k-d}$

- I.e. $a_1g_{n-1} + \cdots + a_dg_{n-d} = \begin{cases} b_0 & n = 0 \\ b_1 & n = 1 \\ \vdots & \\ b_{N-1} & n = N-1 \\ 0 & n \geq N \end{cases}$

Partial Fractions

- Partial fractions must exist for functions that a ratios of polynomials where the numerator has a higher degree than the denominator
- These can be used to deduce a generating series for a more complex function (i.e. a ratio of polynomials) by splitting it into smaller, more manageable series
- Theorem 4.14 (paraphrase): there must be a closed form formula for a given term of a recurrence relation with a generating series that is a ratio of polynomials

Inhomogeneous Linear Recurrence Relations

- Here, $g_n = -(a_1g_{n-1} + a_2g_{n-2} + \cdots + a_dg_{n-d}) - c - dn = 0$ for some $c \neq 0, d \in \mathbb{Z}$
- More general strategy for solving:
  - Re-write as $g_n + a_1g_{n-1} + a_2g_{n-2} + \cdots + a_dg_{n-d} = c + dn$
  - Multiply both side by x^n
  - Take the sum over all of $n \geq 2$ on each side
  - Solve the sums and equate the two sides

Polyexps

- $q : \mathbb{N} \to \mathbb{C}$ is a **polyexp** iff $q(n)$ can be expressed as a sum of polynomials, each with a complex exponent
- A function can be **eventually polyexp** if it is polyexp past a certain point (i.e. if $n \geq N$)

Theorem 4.18

Let g = $(g_0, g_1, g_2, \ldots)$ be a sequence of complex numbers. The following are equivalent

1. The sequence g satisfies a HLRR (with initial conditions)
2. The sequence g satisfies a possibly inhomogeneous linear recurrence relation (with initial conditions) in which the RHS is an eventually polyexp function
3. The generating series $G(x)$ is a rational function

4. The function $g(n) = g_n$ is polexp

Quadratic Recurrence Relations

- A recurrence is **quadratic** iff its generating series $G(x)$ can be expressed as $A(x)G(x)^2 + B(x)G(x) + C(x) = 0$, where $A(x), B(x), C(x)$ are power series
- Using the quadratic formula, we get $G_+, G_- = \dfrac{-B(x) \pm \sqrt{B(x)^2 - 4A(x)C(x)}}{2A(x)}$
  - The correct series is the one without negative coefficients or exponents

Catalan Numbers

- $C_n = \dfrac{1}{n+1}\binom{2n}{n}$
- These count the number of well-formed parenthesizations, binary trees, lattice paths ($n \times n$ grid without crossing $y = x$), ways to partition a convex polygon into triangles, etc

Question Strategies

- Finding a closed form formula for a given term n of a recurrence relation: this is simply the inside of the generating series

Chapter 5 - Introduction to Graph Theory

Graph

A **graph** G is a finite, nonempty set of **vertices** $V(G)$ together with a finite, nonempty set of **edges** $E(G)$, which are *unordered* pairs of distinct vertices.

- Note that edges do not have direction and that two vertices can have at most one edge between them
- **Adjacent** vertices have an edge connecting them. That edge is **incident** to / **joins** both vertices
- **Neighbors** ($N(u)$): set of vertices adjacent to u
- **Planar**: G can be drawn without lines crossing

Isomorphism

Isomorphism

Graphs G_1 and G_2 are isomorphic if there exists a bijection $f : V(G_1) \to V(G_2)$ such that vertices $f(u)$ and $f(v)$ are adjacent in $G_2 \iff$ they are adjacent in G_1

- I.e. the graphs have the same shape (*iso=same, morph=shape*)
- **Isomorphism class**: set of all graphs that are isomorphic to a given graph
- Every graph is isomorphic to itself (**automorphism**)

Degree

Handshaking Lemma

For any graph G, we have $\sum_{v \in V(G)} \deg(v) = 2|E(G)|$

- k**-regular graph**: every vertex in G has degree k
- **Complete graph**: every set of distinct vertices is adjacent (there are $\binom{k}{2}$ edges)

Bipartite Graphs

Bipartite Graph

A graph G that can be partitioned into two sets A and B such that all edges join a vertex from A to B. A *complete* bipartite graph has all edges in A adjacent to all vertices in B.

- n**-cube**: Graph where the vertices are at each position in $\{0,1\}^n$ and edges connect vertices with exactly one differing digit

Specifying Graphs

Name	Description
Adjacency Matrix	Matrix with rows and columns corresponding to vertices; the overlapping entry is 1 if there is an edge and 0 otherwise
Incidence Matrix	Matrix where rows correspond to vertices and columns correspond to edges; if the edge is incident to the vertex, the entry is 1, 0 otherwise
Adjacency List	List of tuples of each vertex and the list of its neighbors

Paths and Cycles

Subgraph

A subgraph of G has a graph whose vertex set is a subset is a subset U of $V(G)$ and whose edge set is a subset of the edges in G with both endpoints in U

- **Spanning subgraph**: The subgraph has all the vertices from the original graph
- **Walk**: sequence of alternating adjacent edges and vertices
  - **Length**: number of edges in the walk
  - **Closed walk**: the walk ends at the same vertex it started at

Path (walk version) and Theorem 4.6.2

A walk where all the vertices are distinct. If there was a walk between vertices, then there is also a path between them

Cycle

A cycle in a graph G is a subgraph of G with n 2-regular vertices, where the vertices form a closed walk

- **Path** (cycle version): The subgraph obtained by removing an edge from a cycle
- If every vertex in G has degree ≥ 2, then G contains a cycle
- The **girth** $g(G)$ of G is the length of the shortest cycle in G.
  - If G has no cycles, $g(G)$ is infinite
- **Hamilton cycle**: a spanning cycle (contains every vertex of the graph)

Equivalence Relations

- A relation $\mathcal{R}$ between sets S and T can be defined as a subset of $S \times T$. We will mostly consider relations on S, i.e. subsets of $S \times S$

Name	Definition
Reflexivity	Each element in S is related to itself
Symmetry	For $a, b \in S$, if a is related to b, then b is related to a
Transitivity	For $a, b, c \in S$, if a, b are related and b, c are related, then a, c are related
Equivalence relation	A relation that is reflexive, symmetric, and transitive

Connectedness

- A graph G is **connected** if there is a path between any two vertices

> **Component**
> A component of G is a subgraph C of G such that C is connected and no subgraph of G that properly contains C is connected (essentially, a part of the graph that is completely disconnected from the rest of it)

- **Cut** of $X \subseteq V(G)$: set of edges in $V(G)$ that have exactly one edge in X
- G is disconnected $\iff$ there exists an $X \subset V(G)$ such that the cut of X is empty

Eulerian Circuits

> **Eulerian Circuit**
>
> A closed walk in G that contains every edge of G exactly once.

- If G is a connected graph, then it has a Eulerian circuit $\iff$ every vertex has an even degree

Bridges

> **Bridge**

An edge $e \in E(G)$ is a bridge if G/e has more components than G

- If e is a bridge, then G/e must have exactly two components, where each endpoint of the bridge is in a different component
- An edge e is a bridge $\iff$ it is not contained in a cycle

Proof strategies

- Most graph theory proofs proceed by contradiction
- Also common: Induction, where the inductive case is applied recursively to neighbors
- When proving directly, consider the contrapositive
- Longest path argument: declare $v_1, v_0, \dots v_k$ to be the longest path in G, proceed directly or by contradiction (by showing that it is not the longest path)
    - Often, show that this leads to a cycle
- Try forming a tree, then use the properties of trees in the proof
- If connectedness is assumed, the path between any two vertices can be used without loss of generality

Chapter 6 - Trees

Trees and Forests

A **tree** is a connected graph with no cycles
A **forest** is a graph with no cycles (connectedness not required)

Properties of Trees

- A unique path exists between any two vertices u and v in a tree T
- Every edge e of a tree T is a bridge
- A tree T with n vertices has $n-1$ edges
  - If a graph G is connected with n vertices and $n-1$ edges, then G is a tree
- A tree with a least two vertices has at least two leaves

Spanning Trees

- A **spanning tree** is a spanning subgraph that is also a tree
- A graph G is connected $\iff$ G has a spanning tree
- Adding an edge to a spanning tree produces a cycle; removing a different edge from this cycle produces another spanning tree
- If T is a spanning tree of G and e is an edge in T, then $T-e$ has two components

Characterizing Bipartite Graphs

- A subgraph of a bipartite graph is also bipartite
- Graph G is bipartite $\iff$ G has no cycles of odd length

Breadth-First Search

- **Parent/predecessor** function: $pr(x) : V(T) \to V(T)$ is the first vertex in the unique path from vertex x to a given vertex u in the tree T. $pr(u)$ is defined as $\varnothing$
- Algorithm for finding a spanning tree of an arbitrary G
  - Select an initial vertex u and define $pr(u) = \varnothing$. This is the initial subgraph D
  - Until D is full, continue adding edges to D that have one vertex r in D and one vertex v outside it. I.e. add vertex v and define $pr(v) = r$.
  - Refinement (breadth-first search): at each stage, choose an edge incident to the unexhausted vertex that joined the tree the earliest

- If $|V(D)| = |V(G)|$ when the algorithm terminates, D is a spanning tree. Otherwise, G is disconnected and no spanning tree exists
- **BFS Search tree**: spanning tree with directed edges from each vertex to its parent; each vertex has a **level**, which is its distance from the **root.** Non-tree (cycle-forming) edges may be present in the graph
  - **Exhausted vertex**: vertex that is not adjacent to a vertex outside the tree
  - Non-tree edges join vertices at most one level apart

Applications of BFS

- A connected graph G with BFS tree T has an odd cycle $\iff$ G has a non-tree edge joining vertices at the same level in T
- The length of a shortest path from u to v in a connected graph G is equal to the level of v in any BFS tree of G with root u

Minimum Spanning Tree

- **Minimum Spanning Tree**: Spanning tree with minimum sum of weighted edges
- *Prim's Algorithm* is a greedy algorithm for finding minimum spanning trees:
  - Let vertex $v \in G$ be arbitrary; let T be the tree that consists of only v. While T is not a spanning tree of G
  - Let $e = uv$ be an edge with the smallest weight in the cut induced by $V(T)$, where $u \in V(T)$ and $v \notin V(T)$. Add e to $E(T)$ and v to $V(T)$

Proof Strategies

- Proving a graph is not bipartite: find a subgraph that is not bipartite (ex. an odd cycle). Then, by contrapositive, the original graph cannot be bipartite either
- Show that a subgraph is a tree (or has a spanning tree), then derive meaning from that

Chapter 7 - Planar Graphs

Planarity

Planarity

A graph is **planar** if it can be drawn on a plane such that no edges cross and no vertices are at the same positions. Such a drawing is a **planar embedding** of the graph

- This separates the graph into **faces**, including the unbounded outside one
  - The edges of a face are its **boundary**
  - Faces that share a boundary are **adjacent**
  - **Degree**: number of edges in the *boundary walk*
    - A boundary will be counted twice when determining degree if it is a bridge
    - If there are multiple components, then the degree is the sum of that of the components

Faceshaking Lemma

The total degree of all the faces in the planar embedding of graph G is $2|E(G)|$, i.e.
$\sum_{i=1}^{s} \deg(f_i) = 2|E(G)|$

Euler's Formula

Euler's Formula

Let G be a connected graph with k components, v vertices and e edges. If G has a planar embedding with f faces, then $v - e + f = k + 1$

- For a connected graph, $v - e + f = 2$

Stereographic Projection

- A graph G is planar $\iff$ G can be drawn on a sphere
- The two graphs can be converted between by *stereographic projection*

Platonic Solids

- **Platonic graph**: G is platonic if it has a planar embedding where each vertex has the same degree $d_v \geq 3$ and each face has the same degree $d_f \geq 3$
- There are exactly 5 platonic graphs, corresponding to the 5 platonic solids

Nonplanar Graphs

- If G is cyclic, then the boundary of each face in the planar embedding of G contains a cycle
- In a planar graph G with $v \geq 3$ vertices and e edges, we have $e \leq 3v - 6$
  - If G is bipartite, then we have $e \leq 2v - 4$

Kuratowski's Theorem

- **Edge subdivision**: replacing an edge with a new path with a length of at least 1

> **Kuratowski's Theorem**
> A graph G is not planar $\iff$ G has a subgraph that is an edge subdivision of K_5 or $K_{3,3}$

- To find this subgraph, look for a long cycle and see if any connections can be made to create an edge subdivision of K_5 or $K_{3,3}$

Coloring and Planar Graphs

- An edge $e \in E(G)$ is **contracted** by "shortening" it such that its ends become one vertex
- **k-coloring of G**: function $f : V(G) \to C$, where C is a set of "colors" of size k
- A graph G is 2-colorable $\iff$ G is bipartite
- K_n is n-colorable, and not k-colorable for any $k < n$
- Every planar graph is 6-colorable (6-color theorem) and 5-colorable (5-color theorem)

> **Four Color Theorem**
> Any planar graph G is 4-colorable

Dual Planar Maps

> **Dual graph**
>
> For a planar graph G, the dual G^* of G is constructed replacing each face of G with a vertex and drawing an edge between each pair of vertices whose corresponding faces were adjacent

- A face with degree k in G becomes a vertex of degree k in G^*
- Thus, the 4-color theorem also applies to coloring the faces of planar graphs

Proof Strategies

- Use handshaking/faceshaking lemma to calculate the number of edges and vertices
- Average number of edges/vertices/etc.: if the average number of these is a, show that $\lfloor a \rfloor$ and $\lceil a \rceil$ edges/vertices/etc. must exist
- Use $e \leq 3v - 6$ and $e \leq 2v - 4$ to constrain the number of edges and vertices in graphs
- Planar graphs will have the following
    - K_5 or $K_{3,3}$ (Kuratowski's theorem)
    - $e \leq 3v - 6$
    - More than one face $\iff$ at least one cycle

Chapter 8 - Matchings

Matching

> **Matching**
>
> A **matching** in a graph G is a set M of G's edges such that no two edges in M have a common end

- A vertex v of G is **saturated** by $M \iff v$ is incident with an edge in M (i.e. v is in one of the edges in M)
- **Maximum matching** of G: the largest possible matching in G
  - **Perfect matching** of G: matching that contains every vertex of G
- **Job Assignment Problem**: finding the maximum matching of a bipartite graph

Paths

- **Alternating Path**: a path in G such that edges are alternately in and not in matching M
- **Augmenting Path**: an alternating path that joins two distinct vertices that aren't saturated by M (these are on each end of the path)
- If M has an augmenting path, then it is not a maximum matching

Covers

> **Cover**
>
> A **cover** of a graph G is a set C of vertices such that every edge of G has at least one end in C

- If M and C are matchings and covers of G respectively, then $|M| \leq |C|$. If $|M| = |C|$, then M is a maximum matching and C is a minimum cover

König's Theorem

> **König's Theorem**
>
> In a bipartite graph, the maximum size of a matching is the minimum size of a cover

Algorithm for maximum matching in bipartite graphs

- G is a graph with bipartitions A and B, and matching M
- From matching M, construct X and Y
- Let $\hat{X} = \{v \in A : v \text{ is unsaturated }\}$ $\hat{Y}$ be $\emptyset$, and $pr(v)$ be undefined for all vertices v
- For each vertex $v \in B - \hat{Y}$ that has an edge (u, v) with $u \in \hat{X}$, add v to $\hat{Y}$ and set $pr(v)$ to u
- Once the last step adds no vertex to $\hat{Y}$, return the maximum matching M and the minimum cover $C = \hat{Y} \cup (A - \hat{X})$
- I.e. if there is an unsaturated vertex v in Y, find an augmenting path $P(v)$ ending at v and use it to construct a larger matching M'. Replace M with M' and reconstruct X and Y

Applications of König's Theorem

- **Neighbor set** $N(D)$ of $D \subseteq V(G)$: all vertices that are incident to at least one vertex in D

> **Hall's Theorem**
> A bipartite graph G with bipartition A, B has a matching saturating every vertex in $A \iff$ every $D \subseteq A$ satisfies $|N(D)| \geq |D|$

- If a given subset $D \subseteq A$ has $|N(D)| < |D|$, then there are more vertices than possible candidates for a member of the matching

Systems of Distinct Representatives

- Divide the population into different (overlapping) groups $Q_1 \ldots Q_n$. What is the best way to pick representatives for each group such that the representative is in the group the represent and no one represents two different groups?
- Can be constructed as a bipartite graph A, B where A is the set of people and B are the set of groups. Then, the optimal system is the maximal matching

> **Hall's SDR Theorem**
>
> The collection $Q_1 \ldots Q_n \subseteq Q$ has an SDR $\iff$ for every subset $J \subseteq \{1, 2, \ldots n\}$, we have $\left|\bigcup_{i \in J} Q_i\right| \geq |J|$

Perfect Matchings in Bipartite Graphs

- A bipartite graph G with bipartition A, B has a perfect matching $\iff |A| = |B|$ and every $D \subseteq A$ satisfies $|N(D)| \geq |D|$
- If G is a k-regular bipartite graph with $k \geq 1$, then G has a perfect matching

Edge-coloring

- **Edge k-coloring** a graph relies on partitioning the graph into k matchings
- A bipartite graph with maximum degree δ has an edge δ-coloring
- Let G be a bipartite graph having at least one edge. Then G has a matching saturating each vertex of maximum degree
- Application to timetabling: coloring the edges of a bipartite graph where A are the activities and B are the people doing the activities
- Bound edge-coloring problem: what is the smallest number of colors needed to edge-color a bipartite graph if no color can be assigned to more than m edges?

Theorem 8.8.1

Let G be a graph with e edges and let $k, m \in \mathbb{N}^+$ such that G has an edge k-coloring and $q \leq km$. Then G has a k-coloring in which every color is used at most m times.

Proof Strategies

- Consider the consequence of all theorems that relate vertices and edges
  - I.e. Handshaking/faceshaking lemmas, Euler's theorem, $|V| = |E| + 1$ for trees, etc.
- To show a perfect matching, assume none, then build augmenting path (for contradiction)
- Convert between cover and matching using $|M| = |C|$ condition
- If M is a maximum matching, then there is no edge with 2 unsaturated vertices, so the set of saturated vertices is a cover
- If a matching M is a maximum matching, then (by contradiction) there is no $v = (a, b) \in V$ such that neither a nor b are unsaturated. Thus, the saturated vertices must form a cover