Discrete Mathematics Quick Guide

Matroids

A pair (E, I) is a matroid if:

1. $\varnothing \in I$ (I is not empty)
2. $B \subset A \wedge A \in I \Rightarrow B \in I$ (I is an independent set).
3. If $A, B \in I$ such that $|B| = |A| + 1$ then $\exists v \in B \setminus A$ s.t. $A \cup \{v\} \in I$ (Matroid property).

S is a **basis** if it is maximal in I. $|S|$ is the rank of the matroid.

Greedy algorithms always yield the min/max in matroids.

Spanning trees are matroids.

Graph Theory

$E = 1/2 \cdot \sum_{v \in V} \delta(v)$	*Handshaking lemma*
Connected & acyclic	*Tree definition*
Maximally acyclic	*Tree characterization*
Minimally connected	*Tree characterization*
$E = V - 1$	*Edges in a tree*
Kruskal's algorithm	*Greedy for min/max spanning tree*
If G planar; $V - E + F = 2$	*Euler's characteristic*
If G planar; $E \leq 3n - 6$	*Edge bound for planar graphs*
Matrix tree theorem	*Counts the number of spanning trees*
Dijkstra's Algorithm	*Shortest path; no negative weights*
Moore's Algorithm (Bellman-Ford)	*Shortest path, detects - cycles*
Floyd-Warshall Algorithm	*Shortest path, weighted adj. matrix*
Ford-Fulkerson Algorithm	*Path finding for max flow*
$\delta(v)$ is even $\forall v \in V$	*Eulerian graph*
$M \subseteq E \,:\, e \cap f = \varnothing \forall e, f \in M$	*Matching*

Max flow - min cut Theorem: The maximum flow corresponds to the weight of a minimum cut.

A graph is **bipartite** if and only if it has no cycles of odd length.

A matching is **perfect** if it covers all vertices.

Hall's Theorem: A bipartite graph A, B admits complete matching in A if and only if for every $S \subseteq A$, $|S| \leq |N(S)|$ where $N(S)$ is the set of vertices of B that are adjacent to S.

Combinatorics

If $A \cap B = \varnothing$, then $\lvert A \cup B\rvert = \lvert A\rvert + \lvert B$	*Sum principle*
$\lvert A \times B\rvert = \lvert A\rvert \times \lvert B\rvert$	*Product principle*
$f : A \to B$ bijective, $\Rightarrow \lvert A\rvert = \lvert B\rvert$	*Bijection principle*
$\lvert A\rvert > \lvert B\rvert \Rightarrow f : A \to B$ is non-injective	*Pigeonhole principle*
$\lvert R\rvert = \sum_{i \in A} \lvert R_{i,0}\rvert = \sum_{j \in B} \lvert R_{0,j}\rvert$	*Double counting*
$\lvert A_1 \cup A_2\rvert = \lvert A_1\rvert + \lvert A_2\rvert - \lvert A_1 \cap A_2\rvert$	*Inclusion/Exclusion principle*
$\lvert\{A \,:\, A \subseteq M\}\rvert = 2^{\lvert M\rvert}$	*Counting subsets of M*
$\lvert\{A \,:\, A \subseteq M \wedge \lvert A\rvert = k\}\rvert = \binom{\lvert M\rvert}{k}$	*Counting subsets of size k*
$\lvert M^k\rvert = \lvert M\rvert^k$	*Counting sequences of length k*
$n!$	*Permutations with n elements*
$\binom{n}{k} = \frac{n!}{(n-k)!k!}$	*Combinations formula*
$\binom{n}{k} = \binom{n}{n-k}$	*Combinations identity*
$s_{n,k} = \frac{n!}{k!}\left(\sum_{a_1+a_2+\ldots+a_k=n} \frac{1}{a_1 \cdot a_2 \cdot \ldots \cdot a_k}\right)$	*First kind Stirling numbers*
$S_{n,k} = \frac{n!}{k!}\left(\sum_{a_1+a_2+\ldots+a_k=n} \frac{1}{a_1! a_2! \ldots a_k!}\right)$	*Second kind Stirling numbers*
$C_n = \frac{1}{n+1}\binom{2n}{n}$	*Catalan numbers*

Catalan numbers count full binary trees with $n+1$ leaves, convex polygons with $n+2$ sides. They are frequently found in binary constructions.

Generalized inclusion/exclusion : $|A_1 \cup \ldots A_n| = |A_1| + \ldots + |A_n| - |A_1 \cap A_2| - \ldots - |A_{n-1} \cap A_n| + |A_1 \cap A_2 \cap A_3| + \ldots$.

Stirling numbers of the first kind: $s_{n,k}$, number of permutations of n elements with exactly k cycle factors.

Stirling numbers of the second kind: $S_{n,k}$, in how many ways can we partition n elements into k non-empty parts

Generating Functions

Let $(a_n), (b_n), (c_n)$ be sequences

$A(x) = \sum_{i=0}^{\infty} a_i x^i$	*Ordinary generating function of* (a_n)
$\hat{A}(x) = \sum_{i=0}^{\infty} \frac{a_i}{i!} x^i$	*Exponential generating function of* (a_n)
$(a_n) = \overline{1} \Leftrightarrow A(x) = \frac{1}{1-x}; \hat{A}(x) = e^x$	*Basic generating functions*
$c_n = a_n + b_n \Leftrightarrow C(x) = A(x) + B(x)$	*OGFs are additive*
$c_n = \alpha a_n \Leftrightarrow C(x) = \alpha A(x)$	*OGFs are scalar*
$c_n = a_n + b_n \Leftrightarrow \hat{C}(x) = A(x) + B(x)$	*EGFs are additive*
$c_n = \alpha a_n \Leftrightarrow \hat{C}(x) = \alpha A(x)$	*EGFs are scalar*

Generating Functions

Let $(a_n), (b_n), (c_n)$ be sequences

$$C(x) = A(x)B(x) \Leftrightarrow c_n = \sum_{k=0}^{n} a_k b_{n-k} \qquad \textit{Cauchy product(OGF)}$$

$$\hat{C}(x) = \hat{A}(x)\hat{B}(x) \Leftrightarrow c_n = \sum_{k=0}^{n} \binom{n}{k} a_k b_{n-k} \qquad \textit{Combinatorial product(EGF)}$$

$$A(x) = \sum_{n=0}^{\infty} \frac{A^{(n)}}{n!} x^n \qquad \textit{Taylor series at } 0$$

$$C(x) = xA(x) + c_0 \Leftrightarrow c_n = a_{n-1} \qquad \textit{Right shift(OGF)}$$

$$C(x) = \tfrac{A(x)-a_0}{x} \Leftrightarrow c_n = a_{n+1} \qquad \textit{Left shift(OGF)}$$

$$C(x) = A'(x) \Leftrightarrow c_n = (n+1)a_{n+1} \qquad \textit{Derivative(OGF)}$$

$$\sum_{n=0}^{\infty} \binom{n+k-1}{k-1} x^n = \frac{1}{(1-x)^n} \qquad \textit{Lemma(OGF)}$$

Generating functions are compact expressions for sequences.

GFs are useful to solve recurrence equations:
Step 1: Express the recurrence in terms of the GFs. **Step 2**: Solve the functional equation. **Step 3:** Express the solution as a power series to obtain the coefficients, and the solution to the recurrence.

GFs are also useful in order to apply the symbolic combinatorial method.

Combinatorial symbolic method

In combinatorics we are interested in counting how many objects of size n exists with certain properties. The properties can be abstracted into a type $\mathcal{A}$ with a size function $w : \mathcal{A} \to \mathbb{N}$.

The combinatorics of type $\mathcal{A}$ is reduced to computing a sequence (a_n), such that a_i corresponds to the ammount of objects of type $\mathcal{A}$ of size i.

The symbolic method allows us to compute the generating function $A(x)$ based only on the construction rules for the type $\mathcal{A}$.

Let $\mathcal{A}, \mathcal{B}, \mathcal{C}$ be combinatorial categories.
Sum type: We define $\mathcal{C} = \mathcal{A} + \mathcal{B} = \mathcal{A} \cup \mathcal{B}$. The weight is the weight for category $\mathcal{A}$ or the weight for category $\mathcal{B}$.
$C(x) = A(x) + B(x)$.

Product type: We define $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ to be the combinatorial category resulting from joining an object of type $\mathcal{A}$ with an object of type $\mathcal{B}$. The weight is given by adding the weight of the part in $\mathcal{A}$ to the weight of the part in $\mathcal{B}$.
$C(x) = A(x)B(x)$

Combinatorial symbolic method

Sequence type: We define $\mathcal{C}$ as a finite sequence of objects of category $\mathcal{A}$. The weight is given by the sum of the weights in the sequence. For example, a rooted plane tree is either a leaf or a root and a sequence of trees.
$C(x) = \dfrac{1}{1 - A(x)}$

Partition type: We define $\mathcal{C} = \mathcal{A} * \mathcal{B}$ as a partition made by an ordered choice of objects from $\mathcal{A}$ and objects from $\mathcal{B}$. The weight is given by adding the weight of the selection from $\mathcal{A}$ to the weight of the selection of $\mathcal{B}$.

$\hat{C}(x) = \hat{A}(x) \times \hat{B}(x)$

Set type: The set type $\mathcal{C} = \mathit{set}(\mathcal{A})$ is given by an unordered collection of objects of type $\mathcal{A}$. The weight is given as the sum of weights in the collection.
$\hat{C}(x) = e^{\hat{A}(x)}$
Cycle type: The cycle type $\mathcal{C} = \mathit{cyc}(\mathcal{A})$ is given by cycles formed by objects of $\mathcal{A}$. The weight is the sum of weights of objects in the cycle.

$$\hat{C}(x) = \log\left(\frac{1}{1 - \hat{A}(x)}\right)$$

Notice that if a combinatorial object can be described built through these operations, then their generating functions is easily obtained by applying the previous rules.

Posets

A **poset** is a pair $(A, \leq)$, A is a set $\leq$ is a relation that is

- **Reflexive:** $\forall x \in A,\ x \leq x$
- **Transitive:** $\forall x, y, z \in A;\ x \leq y \wedge y \leq z \Rightarrow x \leq z$
- **Antisymmetric:** $\forall x, y \in A;\ x \leq y \wedge y \leq x \Rightarrow x = y$

A poset $(A, \leq)$ is **linearly ordered** if $\forall x, y \in A;\ x \leq y \vee y \leq x$.

Let $(A, \leq)$ be a poset, $(B, \leq)$ is a **chain** if $B \subseteq A$, and $(B, \leq)$ is linearly ordered.

An element $x \in (A, \leq)$ is **maximal** if $\forall y \in (A, \leq);\ x \leq y \Rightarrow y = x$

An element $x \in (A, \leq)$ is **minimal** if $\forall y \in (A, \leq);\ x \geq y \Rightarrow y = x$

x is the 1 element of $(A, \leq)$ if $\forall y \in (A, \leq);\ x \geq y$

x is the 0 element of $(A, \leq)$ if $\forall y \in (A, \leq);\ x \leq y$

Posets

A **closed interval** $[a, b]$ of a poset $(A, \leq)$ is defined as $\{c \in A \mid a \leq c \leq b\}$.

A poset is **locally finite** if any closed interval is finite.

A **Hasse diagram** is a visual representation of a locally finite poset $(A, \leq)$. Elements of A are represented by vertices of a graph. If $a \leq b$, then there is a path in the Hasse diagram from a to b, and b has a higher geometrical position than a.

The Kronecker delta function is defined as $\delta(x, y) = \begin{cases} 0 & \text{if } x \neq y \\ 1 & \text{if } x = y \end{cases}$.
The delta function defines implicitly the Möbius function for locally finite posets:

$$\sum_{x \in [a,b]} \mu(x, b) = \delta(a, b)$$

Möbius inversion theorem: Let $(P, \leq)$ be a locally finite poset with a 0-element and μ its Möbius function.
Let $S_f(x) = \sum_{z \in [0,x]} f(z)$.
Then $f(x) = \sum_{z \in [0,x]} S_f(z) \cdot \mu(z, x)$.
Intuitively the Möbius inversion theorem allows us to compute the value of a function in terms of cumulative sums of previous values and the Möbius function of a poset.

A poset $(A, \leq)$ is a **lattice** if for any $a, b \in A$, there exists a minimal upper bound (**join**, $a \vee b$) and a maximal lower bound (**meet**, $a \wedge b$) for a and b.

A lattice L is said to be a **complete lattice** if any non empty subset of L has a meet and a join.

Number Theory

Let $a, b \in \mathbb{Z}$, we say that $a \mid b$ if there exists $c \in \mathbb{Z}$ such that $a \cdot c = b$.

Let $a, b \in \mathbb{Z}$, we say that $d = \gcd(a, b)$ is the **greatest common divisor** if and only if $d \mid a$, $d \mid b$ and for any common divisor c of a, b, then $c \mid d$. For uniqueness we usually take $\gcd \geq 0$.

Integer long division : Let $a, b \in \mathbb{Z}$ such that $b > 0$, then there exist unique $q, r \in \mathbb{Z}$ such that $a = b \cdot q + r$ and $0 \leq r < b$. Notice that r is usually called the remainder, and q the quotient.

Number Theory

Euclidean algorithm: Given $a, b \in \mathbb{Z}$, iteratively apply integer long division in the following way:

$$a = b \cdot q_0 + r_0$$

$$b = r_0 \cdot q_1 + r_1$$

$$r_0 = r_1 \cdot q_2 + r_2$$

$$\vdots$$

$$r_{k-1} = r_k \cdot q_{k+1} + 0$$

Then r_k is the $\gcd(a, b)$ and it can be expressed as a linear combination of a and b.

Let $p \in \mathbb{Z}$, and $p > 1$, p is **prime** if and only if $\pm 1, \pm p$ are the only divisors of p. We denote by $\mathbb{P}$ the set of prime integers, which is infinite.

Prime characterization: $p \in \mathbb{P}$ if and only if $p > 0$ and $p \mid (a \cdot b) \Rightarrow p \mid a \vee p \mid b$.

We say that a and b are **co-prime** if and only if $\gcd(a, b) = 1$.

Fundamental theorem of arithmetics: Let $z \in \mathbb{Z}$, $z > 1$ then there exists a unique factorization up to factor permutation of $z = \prod_{i=1}^{k} p_i^{\alpha_i}$, where all p_i are primes and all $\alpha_i > 0$.

We define the **least common multiple** of a and b as a common multiple that divides any other common multiple. Notice that $\operatorname{lcm}(a, b) \cdot \gcd(a, b) = a \cdot b$.

The gcd and the lcd of a and b can also be defined by their prime factorization. Let $S = \{p \in \mathbb{P} : p \mid a \vee p \mid b\}$. Notice that $a = \prod_{p \in S} p^{\alpha_p}$ and $b = \prod_{p \in S} p^{\beta_p}$ with $0 \leq \alpha_p, \beta_p$. Then $\gcd(a, b) = \prod_{p \in S} p^{\min(\alpha_p, \beta_p)}$, $\operatorname{lcm}(a, b) = \prod_{p \in S} p^{\max(\alpha_p, \beta_p)}$.

We define a **congruence relation** modulo n as follows: Let $a, b, n \in \mathbb{Z}$. We say that $a \underset{n}{\equiv} b$ (a is congruent to b modulo n) if $n \mid a - b$. This is equivalent to $a = n \cdot \alpha + b$ for some $\alpha \in \mathbb{Z}$.

Congruence relations are equivalence relations, therefore for any $n \in \mathbb{Z}$, we can define $\mathbb{Z}_n$ as the equivalence classes via the congruence relation modulo n.

Each $\mathbb{Z}_n$ is a ring with the inherited product and sum from $\mathbb{Z}$. Notice that $\mathbb{Z}_p$ is a field if and only if $p \in \mathbb{P}$.

Number Theory

Chinese remainder theorem: Consider a system of congruence equations : $x \underset{m_1}{\equiv} a_1, x \underset{m_2}{\equiv} a_2, \ldots, x \underset{m_k}{\equiv} a_k$. The system has a unique solution modulo $m_1 \cdot m_2 \cdot \ldots \cdot m_k$ if and only if all of the m_i, are pairwise co-prime.

Euler's totient function: We define $\varphi(m) = |\mathbb{Z}_m^*| = |\{0 < z < m : \gcd(z, m) = 1\}|$. $\varphi(p) = p - 1$ if and only if $p \in \mathbb{P}$.

Let $m = \prod_{i=1}^{k} p_i^{\alpha_i}$ such that $p_i \in \mathbb{P}$ and $0 < \alpha_i$, then $\varphi(m) = m \cdot \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right)$.

Euler-Fermat theorem: If $\gcd(a, m) = 1$ then $a^{\varphi(m)} \underset{m}{\equiv} 1$.

Fermat's little theorem: Let $p \in \mathbb{P}$, then for every $a \in \mathbb{Z}$ such that $p \not| a$ it holds that $a^{p-1} \underset{p}{\equiv} 1$

RSA encryption Let $p, q \in \mathbb{P}$, $m = p \cdot q$ and $v = \operatorname{lcm}(p-1, q-1)$. The **encryption key** (public key) $e \in \mathbb{Z}$ can be chosen as any integer co-prime to v ($\gcd(e, v) = 1$). The **decryption key** (private key) is the product inverse of e modulo v, i.e. $d \cdot e \underset{v}{\equiv} 1$.
To encrypt, simply compute $w^e \bmod m$, where w is the number associated to the symbol that we want to encrypt. To decrypt, compute $c^d \bmod m$, where c is the number that represents the encyrpted symbol. RSA is safe since factorization is computationally hard.

The Carmichael function of a positive integer, $\lambda(n)$ is the smallest positive integer m such that $a^m \underset{n}{\equiv} 1$, for every a co-prime to n. It can also be defined as the maximum order of an element in $\mathbb{Z}_n^*$ (the product group of $\mathbb{Z}_n$). The order of an element $a \in \mathbb{Z}_n^*$ is the smallest k such that $a^k = 1$.

The Carmichael function $\lambda(n)$ can be characterized though the following list of values:

- $\lambda(1) = 1, \lambda(2) = 1, \lambda(4) = 3$
- $\lambda(2^e) = 2^{e-2}$ for $e \geq 2$
- $\lambda(p^e) = p^{e-1}(p-1)$ for any $2 \neq p \in \mathbb{P}$
- $\lambda\left(\prod_{i=1}^{k} p_i^{\alpha_i}\right) = \operatorname{lcm}(\lambda(p_1^{\alpha_1}), \ldots, \lambda(p_k^{\alpha_k}))$

Abstract Algebra

Abstract algebra constructions are meant to generalize numbers and operations in general. A large amount of theorems and results for $\mathbb{Z}$ are still valid if we require only some of their more basic structure.

Group: