DW1: Primes & Eucl. Algorithm

Definition 1.1. *$a, b \in \mathbb{Z}, a \neq 0$, then $a \mid b$ (a divides b) $\Leftrightarrow \exists n \in \mathbb{Z}$ with $b = n \cdot a$.*

Theorem 1.2.
1. *$a \mid b$ then $a \mid bx$ for all $x \in \mathbb{Z}$*
2. *$x = y + z$ and a divides* two *of x, y, z then $a \mid x$, $a \mid y$, and $a \mid z$.*
3. *$a \mid b$ and $a \mid c$ then $a \mid (bx + cy)$ for all $x, y \in \mathbb{Z}$*

Theorem 1.3. *$a, b \in \mathbb{Z}, b > 0$, then there exist* unique *$q, r \in \mathbb{Z}, 0 \leq r < b$, with*

$$a = qb + r.$$

Theorem 1.4. *For all $a, b \in \mathbb{Z}$, with $a \neq 0$ or $b \neq 0$, there is a* unique $\gcd(a, b)$.

Theorem 1.5 (Bézout Identity).

$$\gcd(a, b) = \min\{ax + by > 0 \mid x, y \in \mathbb{Z}\}$$

Definition 1.6. *$a, b \in \mathbb{Z}$ are* coprime *(or* relatively prime*)* $\Leftrightarrow \gcd(a, b) = 1$

Theorem 1.7 (Lamé 1844). *The number of iterations in the Euclidean algorithm is $\in O(\log b)$.*

Theorem 1.8. *for all $a, b \in \mathbb{Z}$, we have $a \cdot b = \gcd(a, b) \cdot lcm(a, b)$.*

Definition 1.9. *$p \in \mathbb{Z}, p > 1$ is* prime *if it has only the two divisors 1 and p, otherwise p is called* composite.

Lemma 1.10. *For all $n \in \mathbb{Z}, n > 1$, there is a prime p with $p \mid n$.*

Lemma 1.11 (Euclid's Lemma). *$a, b \in \mathbb{Z}, p$ prime, $p \mid ab$. Then $p \mid a$ or $p \mid b$ (or both).*

Theorem 1.12 (Fundamental Theorem of Arithmetic). *$n \in \mathbb{Z}, n > 1$, has a* unique *prime factorization (up to reordering).*

DW2: Graph Theory Basics

Definition 2.1. Neighborhood: $N(v) = \{w \neq v | \{v, w\} \in E\}$

Theorem 2.2. *In any $G = (V, E)$, the number of vertices of odd degree is even.*

Definition 2.3. *$G = (V, E)$ is* connected *if there is a path between any two vertices.*

Theorem 2.4. *If a graph $G = (V, E)$ is connected, then $|E| \geq |V| - 1$.*

Theorem 2.5. *If a graph $G = (V, E)$ is acyclic (a forest), then $|E| \leq |V| - 1$.*

Definition 2.6. *We call $G = (V, E)$ a* tree *if it is acyclic and connected.*

Theorem 2.7. *The following are equivalent for a graph $G = (V, E)$*

1. *G is a tree*
2. *G is acyclic and $|E| = |V| - 1$*
3. *G is connected and $|E| = |V| - 1$*
4. *There is a unique path between any two vertices $u, v \in V$*
5. *G is acyclic and adding any $e \notin E$ yields exactly one cycle*

Definition 2.8. *A (multi) graph is called* Eulerian *if there exists a* circuit *that visits every edge exactly once. The corresponding circuit is called* Euler tour.

Theorem 2.9. *A connected graph $G = (V, E)$ is Eulerian if and only if it is connected and each vertex has even degree.*

Theorem 2.10. *Graph search can be implemented in $O(|V| + |E|)$ time.*

Theorem 2.11. *A (directed) graph $G = (V, E)$ is (strongly) connected if there is a (directed) path between any two vertices.*

DW3: Computing Shortest Paths

Definition 3.1. *For fixed start vertex s deefine distance function $dist : V \to \mathbb{R}$ by :*

$$dist(v) = \inf\{c(P) | P \text{ is a walk from } s \text{ to } v\}$$

Function dist must satisfy the triangle inequality*:*

$$dist(v) \leq dist(u) + c(u, v), \forall (u, v) \in E.$$

Observation 3.2. *Subpaths of shortest paths are shortest paths, too.*

Lemma 3.3. *Let $d(v), v \in V$ be* arbitrary vertex labels *such that $d(v) \geq dist(v)$. There holds $d(v) = dist(v)$* if and only if*, for all arcs $(u, v) \in E$, the triangle inequality holds.*

DW4: Maximum Flows & Minimum Cuts

Definition 4.1. *An (s, t)-*flow *in graph $G = (V, E)$ is a function $f : E \to \mathbb{R}$ such that the following are satisfied:*

- *Capacity constraint: for all $(u, v) \in E : 0 \leq f(u, v) \leq c(u, v)$*
- *Flow conservation: for all $u \in V \setminus \{s, t\}$:*

$$\sum_{v:(u,v)\in E} f(u, v) - \sum_{v:(v,u)\in E} f(v, u) = 0.$$

Definition 4.2. *The* value $\text{val}(f)$ *of a flow f refers to the total net flow out of s:*

$$val(f) = \sum_{v:(s,v)\in E} f(s, v) - \sum_{v:(v,s)\in E} f(v, s).$$

Lemma 4.3. *Let f be a flow in G, g be a feasibleflow in the residula graph G_f. Then flow $h = f + g$ is a (feasible) flow in G with value $val(h) = val(f) + val(g)$.*

Definition 4.4. *An* augmenting path *is an (s, t)-path P in the residual graph G_f. The* residual capacity *$r_f(P)$ of P is defined as:*

$$r_f(P) = \min\{r_f(u, v) \mid (u, v) \in P\}.$$

Theorem 4.5. *Given network $G = (V, E, c)$, flow f is maximum flow in G if and only if G_f has no augmenting path.*

Definition 4.6. *An* (s,t)-cut *is a partition* (S,T) *of* V*, such that* $s \in S, t \in T$*. The* arcs in the cut (S,T) *are those with tail in* S *and head in* T*. The* capacity $c(S,T)$ of a cut (S,T) *is* $c(S,T) = \sum_{e \in (S,T)} c(e)$.

Theorem 4.7 (Weak Duality Theorem). *The value of any flow is at most equal to the capacity of any cut. That is, for any flow* f *and any cut* (S,T)*,* $val(f) \leq c(S,T)$.

Theorem 4.8 (Strong Duality Theorem). *There is a flow* f *and a cut* $c(S,T)$ *such that* $val(f) = c(S,T)$.

DW5: Stable Matchings

Definition 5.1. *Given a graph* $G = (V,E)$*, a* matching $M \subseteq E$ *in* G *is a set of edges such that no two edges* $e, e' \in M$ *share a common endpoint. Matching* M *is a* perfect matching *if every* $v \in V$ *is incident to* exactly *one edge in* M.

Definition 5.2. *A graph* $G = (V,E)$ *is* bipartite *if* V *can be partitioned into sets* V_N *and* V_W *such that no edge in* E *has both endpoints in* V_N *and no edge has both endpoints in* V_W.

Definition 5.3. *Given a matching* M*, a pair* (m,w) *of a man and a woman is an* unstable pair *in* M*, if* m *prefers* w *over his matching partner in* M *and* w *prefers* m *over her matching partner in* M*. A bipartite matching* M *is a* stable matching *if it is a perfect matching and there is no unstable pair in* M.

Theorem 5.4. *The Gale-Shapley algorithm terminates after at most* n^2 *many iterations, where* n *is the number of vertices in each side of the bipartition, and returns a stable matching.*

Theorem 5.5. *The men-proposing variant of G-S is man-optimal: every man is matched to the best partner possible in any stable matching. The men-proposing variant of G-S is woman-pessimal: every woman is matched to the worst partner possible in any stable matching* M.

DW6: Recurrence Relations

Theorem 6.1. *There exists a unique solution* $a_n = g(n), n \geq 0$ *to the second order linear recurrence relation* $a_{n+2} + C_1 a_{n+1} + C_2 a_n = f(n), (n \geq 0)$ *with given values* a_0, a_1*, function* $f(n)$ *and constants* C_1, C_2.

Theorem 6.2. *Consider homogeneous relation* $a_{n+2} + C_1 a_{n+1} + C_2 a_n = 0$*. Then, series* $a_n = \lambda^n (\lambda \neq 0)$ *is a solution if and only if* λ is a root *of* characteristic polynomial $x^2 + C_1 x + C_2 = 0$.

De Moivre

$$(\cos\alpha + i\sin\alpha)^n = \cos(n\alpha) + i\sin(n\alpha)$$

Table 10.2 from the book

	$a_n^{(p)}$
c, a constant	A, a constant
n	$A_1 n + A_0$
n^2	$A_2 n^2 + A_1 n + A_0$
$n^t, t \in \mathbb{Z}^+$	$A_t n^t + A_{t-1} n^{t-1} + \cdots + A_1 n + A_0$
$r^n, r \in \mathbb{R}$	Ar^n
$\sin\theta n$	$A\sin\theta n + B\cos\theta n$
$\cos\theta n$	$A\sin\theta n + B\cos\theta n$
$n^t r^n$	$r^n(A_t n^t + A_{t-1} n^{t-1} + \cdots + A_1 n + A_0)$
$r^n \sin\theta n$	$Ar^n \sin\theta n + Br^n \cos\theta n$
$r^n \cos\theta n$	$Ar^n \sin\theta n + Br^n \cos\theta n$

DW7: The Algorithmic Side of RSA Public Key Encription

Lemma 7.1. *Let* $p_1, p_2, \ldots p_k$ *be all prime divisors of* n*, then*

$$\phi(n) = n\Pi_{i=1}^{k}\left(1 - \frac{1}{p_i}\right)$$

Observation 7.2.

- $\phi(p) = p - 1$ *for prime* p
- $\phi(pq) = (p-1)(q-1)$ *for primes* p, q

Theorem 7.3. $(U_n, \cdot)$ *is a group of order* $\phi(n)$.

Theorem 7.4 (Euler's Theorem). *Let* $n > 1$*,* $r = \phi(n)$*,* $M \in \mathbb{Z}$*. If* $\gcd(M,n) = 1$*, then* $M^r \equiv 1 (\mod n)$

RSA outline

Alice:

- *chooses two (large) primes* p, q*,* $n = p \cdot q$
- *computes* $r = |U_n| = \phi(n) = (p-1)(q-1)$
- *chooses exponent* $e < r$ *with* $\gcd(e,r) = 1$
- *computes* $d = e^{-1}$ *in* $\mathbb{Z}_r$*,* $(ed \equiv 1 (\mod r))$
- *publishes* n *and* e

Bob:

- *compute ciphertext* $C = M^e (\mod n)$*, email* C *to Alice*

Alice:

- *decrypt* C *by computing* $C^d \equiv M (\mod n)$