Cheat Sheet

Master Theorem. Suppose the function T satisfies the recurrence relation:

$$T(n) = aT(n/b) + f(n).$$

Define $c = \log_b a$. Then the following hold:

1. If $f(n) = O(n^d)$ for $d < c$ then $T(n) = O(n^c)$
2. If $f(n) = \Theta(n^c \log^k n)$ then $T(n) = O(n^c \log^{k+1} n)$
3. If $f(n) = \Omega(n^d)$ for $d > c$, then $T(n) = O(f(n))$

Polynomial Time Verifier. Given a decision problem A, a *verifier* for A is a polynomial time algorithm $\mathrm{verify}(X, C)$ that takes as input

- an instance X of A, and
- a certificate C (size polynomial in size of X)

and returns a value "accept" or "reject," subject to two conditions:

1. *completeness* if X is a "yes" instance, then there exists a certificate C such that $\mathrm{verify}(X, C)$ returns "accept"
2. *soundness* if X is a "no" instance, then for every certificate C, $\mathrm{verify}(X, C)$ returns "reject"

NP Complete Problems. The following problems are NP complete

1. **3-SAT**
   - *input:* 3-CNF formula $\varphi(x_1, x_2, \ldots, x_n)$, i.e., a Boolean formula of the form $C_1 \wedge C_2 \wedge \cdots \wedge C_k$ where each clause C_i is the disjunction of three literals: $C_i = (x \vee y \vee z)$.
   - *output:* "yes" if φ has a satisfying assignment (i.e., a setting of the variables $x_1, x_2, \ldots, x_n$ such that φ evaluates to "true"), and "no" otherwise.
2. **IS** (independent set)
   - *input:* a graph $G = (V, E)$ and a positive integer k
   - *output:* "yes" if V has an independent set of size (at least) k, and "no" otherwise. Recall that an independent set is a set of vertices I such that no two vertices in I share an edge.
3. **VC** (vertex cover)
   - *input:* a graph $G = (V, E)$ and a positive integer k
   - *output:* "yes" if V has a vertex cover of size (at most) k, and "no" otherwise. Recall that a vertex cover is a set of vertices C such that every edge $e = (u, v) \in E$ has at least one endpoint in C.