Reference Sheet

Unless explicitly stated otherwise, you may use any algorithm discussed in this class or 332 to solve a problem. In particular, you may use any of these functions as libraries (this list is not exhaustive).

Graph Search

- `TwoColor`(G) returns `True` if G can be 2-colored (i.e., is bipartite), `False` otherwise. Running time $\Theta(m+n)$
- `ConnectedComponents`(G) finds the connected components of an undirected graph G. You may assume you get any reasonable representation of this information. Running time $\Theta(m+n)$
- `StronglyConnectedComponents`(G) finds the strongly connected components of a directed graph G. You may assume you get any reasonable representation of the information. Running time $\Theta(m+n)$
- `TopologicalSort`(G) returns a `list` of vertices of a directed graph G in topological order, or `null` if the graph has a cycle. Running time $\Theta(m+n)$
- `CondensationGraph`(G) returns the condensation of a directed graph G (a.k.a., the "meta-graph" of G or "graph of SCCs of G). Running time $\Theta(m+n)$
- `Dijkstra`(G, s, t) finds the length of the shortest path from s to t in a (non-negative) weighted, directed graph G. If t is `null`, the distances from s are stored in every vertex. Running time $\Theta(m + n \log n)$
- `Prims`(G) finds the minimum spanning tree of a (weighted, undirected) graph G. Running time $\Theta(m \log n)$

Arrays

- `QuickSelect`(A, k) returns the value which would be at index k of A if A were sorted. Running time $\Theta(n)$
- `MaxSubarraySum`(A) returns the sum of the maximum sum (contiguous) subarray of A. Running time $\Theta(n)$
- `MergeSort`(A) returns the sorted version of A. Running time $\Theta(n \log n)$

Others

- `GaleShapley(riderPrefs, horsePrefs)` returns the rider-optimal stable matching.
  Running time $\Theta(n^2)$ for n riders and n horses
- `2dClosestPoints`(A) returns the distance between the two closest points of A (where A contains vectors in $\mathbb{R}^2$). Running time $\Theta(n \log n)$
- `EditDistance`(x, y) returns the edit distance between strings x and y.
  Running time $\Theta(mn)$ for strings of length m, n

There's more information on the back!

Other information

Master Theorem For a recurrence of the following form, where a, b, c, d are constants

$$T(n) = \begin{cases} d & \text{if } n \text{ is at most some constant} \\ aT\left(\frac{n}{b}\right) + f(n) & \text{otherwise} \end{cases}$$

Where $f(n)$ is $\Theta\left(n^c \cdot \log^k(n)\right)$ for $k \geq 0, a \in \mathbb{Z}^+, c \geq 1$

- If $\log_b(a) < c$ then $T(n) \in \Theta\left(n^c \cdot \log^k(n)\right)$
- If $\log_b(a) = c$ then $T(n) \in \Theta\left(n^c \cdot \log^{k+1}(n)\right)$
- If $\log_b(a) > c$ then $T(n) \in \Theta\left(n^{\log_b(a)}\right)$