Final Exam: 601.433/633 Introduction to Algorithms (Fall 2024)

Tuesday, December 17, 2024

Name:

Ethics Statement

(If you write solutions on other paper, please copy and sign this statement)

I agree to complete this exam without unauthorized assistance from any person, materials, or device.

Signature: Date:

1 Asymptotics and Recurrences (24 points)

(a) (12 points) Take the following list of functions and arrange them in ascending order of growth rate. That is, if function $g(n)$ immediately follows $f(n)$ in your list, then it should be the case that $f(n)$ is $O(g(n))$. No proofs necessary.

$$
\begin{aligned}
f_1(n) &= \log^* n \\
f_2(n) &= n^{4/3} \\
f_3(n) &= n(\log n)^6 \\
f_4(n) &= 2^{2^n} \\
f_5(n) &= 2^{n^2} \\
f_6(n) &= (\log n)^3
\end{aligned}
$$

(b) Solve the following recurrence relations (asymptotic notation OK). For all of them, you may assume that $T(1) = 1$. You do not need to give a proof or justification.

(i) (6 points) $T(n) = 4T(n/3) + n^2$

(ii) (6 points) $T(n) = T(n-1) + \log n$

2 True/False (24 points)

(a) The expected running time of randomized quicksort on an arbitrary permutation of $\{1, 2, \ldots, n\}$ is equal to the expected running time of basic quicksort (use the first element as the pivot) on a *uniformly random* permutation of $\{1, 2, \ldots, n\}$.
true false

(b) The BPFRT algorithm still runs in $O(1)$ time if it is modified to use groups of size 3 rather than 5.
true false

(c) There is an algorithm that sorts n integers each with k bits in $O(nk)$ time.
true false

(d) The sorting lower bound applies to strings, but not to integers.
true false

(e) There is a bijection between red-black trees and 2-3-4 trees for the same set of keys.
true false

(f) If we use both path compression and union by rank in our Union-Find data structure, then the amortized cost of both Unions and Finds is $\Omega(\log^* n)$.
true false

(g) If we know a set of keys S ahead of time, then it is possible to create a dictionary data structure with lookups that take $O(1)$ time.
true false

(h) Dynamic programming algorithms require the subproblems to have the *optimal substructure property*.
true false

(i) Let I_1 and I_2 be two different maximal independent sets in a matroid. Then $|I_1| = |I_2|$.
true false

(j) The simplex algorithm solves linear programming in polynomial time.
true false

(k) There is a deterministic 3/2-competitive algorithm for the ski rental problem.
true false

(l) In online learning, if the best expert makes $O(1)$ mistakes, then the Weighted Majority algorithm makes $O(\log n)$ mistakes.
true false

3 Amortized Analysis (24 points)

Recall the binary counter example from class: we have a binary counter stored in a binary array A, flipping any bit costs 1, and we do n increments. We proved that the worst-case time for an increment is $\Theta(\log n)$, but that the amortized cost of an increment was at most $2 = \Theta(1)$.

(a) (12 points) Suppose we use the potential function $\Phi = \frac{1}{2}(\# \text{ 1's in } A)$. If we do an increment that causes k bit flips, what is the amortized cost under this potential? Give your solution as both a precise function of k and as an asymptotic function of n (for the worst-case k), and justify your answer.

(b) (12 points) Suppose we use the potential function $\Phi = 2(\#\ 1\text{'s in } A)$. If we do an increment that causes k bit flips, what is the amortized cost under this potential? Give your solution as both a precise function of k and as an asymptotic function of n (for the worst-case k), and justify your answer.

4 Basic Graph Algorithms

A directed graph $G = (V, E)$ is *partially connected* if, for all $u, v \in V$, either u is reachable from v or v is reachable from u (or both).

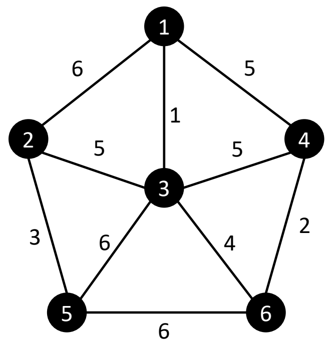
(a) (8 points) Give an example of a directed acyclic graph with a unique source (node with in-degree 0) that is *not* partially connected.

(b) (8 points) Suppose that G is a directed acyclic graph (DAG). Let $v_1, v_2, \ldots, v_n$ be a topological sort of G. Prove that G is partially connected if and only if $(v_i, v_{i+1}) \in E$ for all $i \in [n-1]$.

(c) (8 points) Using the previous part, design an $O(n + m)$-time algorithm to decide if a given DAG G is partially connected. Prove running time and correctness.

5 MST (24 points)

Consider the following weighted graph (numbers inside the nodes are node names, while numbers next to edges denote the weight of the edge):



(a) (12 points) Suppose that we start Prim's algorithm using node 2 as the starting vertex. List the edges in the spanning tree that Prim's algorithm finds in the order in which it adds them (no proof or justification necessary).

(b) (12 points) Now suppose that we run Kruskal's algorithm on the same graph. You can break ties as you see fit. List the edges in the spanning tree that Kruskal's algorithm finds in the order in which it adds them (no proof or justification necessary).

6 Shortest Paths (24 points)

Let $G = (V, E)$ be a directed graph with weighted edges $w : E \to \mathbb{R}$; edge weights can be positive or negative (but not zero), but you are guaranteed that there are no negative-length cycles. Use Johnson's algorithm to design an $O(mn + n^2 \log n)$ time algorithm which determines if G has any cycles of length *exactly* zero (if such a cycle exists then it returns YES, and if no such cycle exists then it returns NO). Justify correctness and running time, but you do not need to give formal proofs.

(Note: there are actually $O(m + n)$ time algorithms for this problem, but they are a bit more subtle. Do not construct such an algorithm for this problem – your algorithm must use Johnson's algorithm.)

7 Max-Flow Min-Cut (24 points)

Let $G = (V, E)$ be a *bipartite* graph. Let $b : V \to \mathbb{N}$. A *multi-matching* is a collection of edges $M \subseteq E$ so every vertex v is incident on at most $b(v)$ edges of M. For example, if $b(v) = 1$ for all $v \in V$, then a multi-matching is actually a traditional matching (as defined in class).

Given bipartite G and b, show how to create a flow network so that the maximum flow is equal to the size of the maximum multi-matching in G. Justify your answer (but formal proof not necessary).

8 NP-Completeness (24 points)

For each of the following statements, say whether they are true or false and justify your answer.

(a) (8 points) True or false: CLIQUE $\leq_p$ 3-SAT.

(b) (8 points) 2-SAT is the same as 3-SAT except that every clause has at most 2 literals. It is well-known that 2-SAT is in P. True or false: if P $\neq$ NP, then 3-SAT $\leq_p$ 2-SAT.

(c) (8 points) True or False: If $\mathrm{P} \neq \mathrm{NP}$, then *no* NP-complete problem can be solved in polynomial time.