"CS 374": Algorithms and Models of Computation, Fall 2015

Midterm 2 – November 9, 2015

Name:	
NetID:	

Section	A	B	C	D	E	F	G	H	I	J
Time	9am	10am	11am	12 noon	1pm	1pm	2pm	2pm	3pm	3pm
Room	1304	1304	1304	1304	1304	1105	1304	1105	1304	1105

#	1	2	3	4	5	Total
Max	15	15	10	10	10	60
Score						
Grader						

- Please *clearly PRINT* your name and your NetID in the boxes above.
- *CIRCLE YOUR SECTION*
- This is a closed-book, closed-notes, closed-electronics exam. If you brought anything except your writing implements, put it away for the duration of the exam. In particular, you may not use *any* electronic devices.
- **Please read the entire exam before writing anything.** Please ask for clarification if any question is unclear.
- There are five problems. Not all problems have the same points.
- If you use Dynamic Programming for any problem recall the rubric from homework. You will get zero points if there is no English description of the recursive function that your algorithm is based on.
- **You have 120 minutes.**
- If you run out of space for an answer, continue on the back of the page, or on the blank page at the end of this booklet, **but please tell us where to look.**
- **Proofs are required only if we specifically ask for them. This policy applies even for greedy algorithms.**

---

1 Short Questions (15 pts)

Part (a) (5 pts) Give an asymptotically tight solution to the following recurrence.

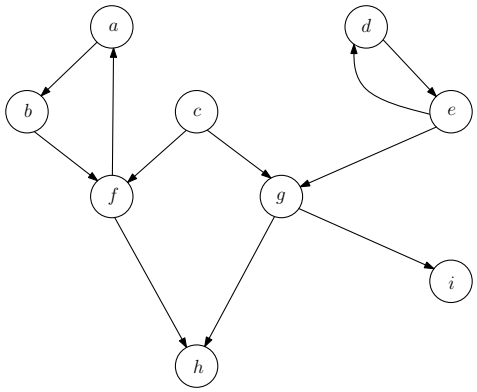
$T(n) = T(7n/8) + T(n/8) + n$ for $n > 8$ and $T(n) = 1$ for $1 \leq n \leq 8$.

Part (b) (5 pts) Let $G = (V, E)$ be an undirected graph with non-negative edge weights given by $w(e)$, $e \in E$. Consider the following generalization of the MST problem. The input consists of G, a node $s \in V$ and an integer k. The goal is to find a minimum weight tree that contains s and exactly k nodes (including s). A natural algorithm that comes to mind is Prim's algorithm that starts with s as the tree and greedily adds nodes until the tree has exactly k nodes. We know that the algorithm works when $k = n$ to give the MST and also works when $k = 1$ (in which case it only outputs s). Prove via a counter example that Prim's algorithm may fail in general for other values of k. You should specify a small graph with edge weights, s and k and show the optimum solution and the output of running Prim's algorithm when stopped with k nodes.

Part (c) (5 pts) Suppose you are given an array A of n numbers and are told that it has very few distinct numbers, say k. Describe an efficient algorithm that given A and k decides whether A has at most k distinct numbers or more than k. A simple algorithm for this is to sort A in $O(n \log n)$ time and count the number of distinct numbers. However we want a faster algorithm when k is quite small. Describe an algorithm with running time $O(n \log k)$. An $O(nk)$ algorithm with get you 3 pts. *Hint:* Use an appropriate data structure.

2 Incomparable Pairs (15 pts)

Let $G = (V, E)$ be a directed graph. We say that a pair of nodes $u, v \in V$ are incomparable if neither u can reach v nor v can reach u. In the example below c, d are incomparable pair and so are f, g. Also, a, c are *not* an incomparable pair since c can reach a.



Part (a) (6 pts)
Describe a linear time algorithm that given $G = (V, E)$ and a node $u \in V$ finds all nodes $v \in V$ such that u, v are incomparable. For example in the above graph if you are given a then the algorithm should output g, d, e, i.

Problem continues on next page.

Part (b) (7 pts)
Describe a *linear-time* algorithm that given a DAG G checks whether there is *any* incomparable pair, and if there is, outputs one of them.

Part (c) (2 pts)
Assuming a linear-time algorithm for DAGs from the previous part describe a linear-time algorithm that given an arbitrary directed graph G checks whether there is any incomparable pair, and if there is, outputs one of them.

3 Shortest Walk (10 pts)

Let $G = (V, E)$ be directed graph with non-negative edge lengths where $\ell(e)$ denotes the length of edge $e \in E$ that represents travel distances in a network. You wish to drive in a car from a start node s to a destination node t while picking up some ice cream at a grocery store and also filling up your car's gas tank. Let $X \subset V$ and $Y \subset V$ be the set of nodes representing the grocery stores and gas stations respectively. Assume that $X \cap Y = \emptyset$, that is, no node in the network represents the location of both a grocery store and a gas station. Moreover, assume that s and t do not represent a grocery store or a gas station. Describe an efficient algorithm that finds the length of a shortest walk from s to t that visits at least one grocery store and at least one gas station. Note that the order in which you visit them is not important. A faster algorithm will get you more points.

4 Closest Points (10 pts)

Let $P = \{p_1, p_2, \ldots, p_n\}$ be n points on the real line given by their coordinates $x_1, x_2, \ldots, x_n$; these are given in an **unsorted** array $A[1..n]$. Let p be another point with coordinate z. Given P, p, their coordinates, and an integer k where $1 \leq k \leq n$ describe an efficient algorithm that outputs the k closest points in P to p. Note the distance between p_i and p is given by $|x_i - z|$. For example if the coordinates of the points are $100, 1, 3, 11, 4, 2, 20, 15, 11$ and $z = 3.1$ and $k = 4$ then the coordinates of the closest k points to p are $1, 2, 3, 4$. For full credit your algorithm should run in $O(n)$ time.

5 Burgers (10 pts)

The McKing chain wants to open several restaurants along Red street in Shampoo-Banana. The possible locations are at $L_1, L_2, \ldots, L_n$ where L_i is at distance m_i meters from the start of Red street. Assume that the street is a straight line and the locations are in increasing order of distance from the starting point (thus $0 \leq m_1 < m_2 < \ldots < m_n$). McKing has collected some data indicating that opening a restaurant at location L_i will yield a profit of p_i independent of where the other restaurants are located. However, the city of Shampoo-Banana has a zoning law which requires that any two McKing locations should be D or more meters apart. Describe an algorithm that McKing can use to figure out the maximum profit it can obtain by opening restaurants while satisfying the city's zoning law.

This page for extra work.