Candidacy (Practice) Exam
Data Structures and Algorithms
Spring 2019

Department of Computer Science and Engineering
The Pennsylvania State University

- Using books, calculators, phones and computers is not allowed during the exam.
- You have 3 hours to solve all six problems.

1 Asymptotic Growth

Insert "T" (true) or "F" (false) in the following table. The statements that are true/false in the respective columns are "$f(n) =$ column header".

$f(n)$	$g(n)$	$O(g(n))$	$\Omega(g(n))$	$\Theta(g(n))$
$\max\{k : k^k \le n\}$	$\log_2 n / \log_2 \log_2 n$			
$n^{\log_{10} n}$	$n^{\log_2 n}$			
$1 + (1 + (-1)^n)\, n$	$1 + (1 - (-1)^n)\, n$			
n^2	$n^3 / (\sqrt{n} \log n)$			
$\log(n!)$	$\log\left(\sqrt{n}^n\right)$			
$n + n \log_2 n$	$n + n \log_{10} n$			

2 Data Structures

Design a data structure for a dynamic set of elements that supports `Insert`, `Delete`, and `Rank` operations in time $O(\log n)$, as well as computing the lower `Median` in constant time. If the dynamic set contains elements with the keys $a_1, a_2, \ldots, a_n$ with $a_1 < a_2 < \ldots < a_n$, then the rank of a_j is j and the lower `Median` would be $a_{\lfloor \frac{n}{2} \rfloor}$. You may assume that all keys are distinct.

You may also assume that you are given a balanced search tree for the dynamic set that handles `Insert` and `Delete` operations. Thus it is sufficient to describe the how to augment it to be able to handle the `Rank` and `Median` operations in the desired running times. Argue that your construction is correct and analyze the time complexity.

3 Graph Algorithms

There are n bus stops in a city numbered 0 to $n-1$. You are given m bus routes (i, j, c_{ij}), where i and j are stops and $c_{ij} > 0$ is the cost of the route between i and j. You may assume that if there is a bus going from i to j, there is also one going in the opposite direction from j to i with the same cost. You are given a starting bus stop s and destination bus stop t. Your task is to find the route from s to t that minimizes your cost.

(a) What algorithm would you use to solve this problem? What is its running time?

Suppose that you are also given k coupons. If you choose to use a coupon in a route, then the price of that particular route will be halved. You are not allowed to use more than one coupon per route, and you are not required to use every coupon.

(b) Show with an example that if you take the optimal $s - t$ path and optimally use your k coupons along this route, there could be better solution; that is, another route that after using the k coupons yields a cheaper cost.

(c) How can you modify the graph so that you can use the algorithm from part (a)? What is the running time of the algorithm in this case?

4 Dynamic Programming

The input describes a set of n cities with $m > n$ one-way connections from one city to another. The connections represent airlines routes: if we have cities a, b and c, and connections $a \to b$ and $b \to c$, then we can flight from a to b and from b to c. We assume that there are no cycles: if $a \to b$ and $b \to c$, then c cannot be connected to a. For every connection $a \to b$ there is a price $p_{a \to b}$ that is given as part of the input.

(a) Design an algorithm to find a linear (or topological) ordering $c_1, c_2, \ldots, c_n$ of the cities such that for every connection $c_i \to c_j$, the starting city c_i appears before in this ordering that the destination city c_j; i.e., $i < j$.

(b) Using the linear ordering from part (a), design an efficient dynamic programming algorithm to compute the cost of the optimal path from a city s to a city t. The running time of your algorithm should be $O(n+m)$. Note that standard algorithms for computing shortest paths would take longer than $O(n+m)$.

5 Divide & Conquer

The maximum (or minimum) value in an array $A[1..n]$ with n elements can be found using only $n - 1$ comparisons. If we are interested in finding both extreme values (the maximum and minimum values), with $2n - 2$ compassions we can accomplish the goal, but this is not optimal.

(a) Consider using a divide-and-conquer approach instead. Derive the recurrence relation for the number of $f(n)$ of comparisons used by this method in the worst case, and write a table showing the values of $f(n)$ for $n = 2, 3, \ldots, 10$.

(b) The optimal number of comparisons needed to find the extreme values in the worst case is $\lceil \frac{3n}{2} \rceil - 2$. Which values in your table are not optimal.

(c) Describe a non-recursive algorithm for this problem that uses the optimal number of comparisons in the worst case. Please keep your description at a high level; do not write a detailed computer program.

6 Greedy Algorithms

A server has n costumers (numbered 0 to $n - 1$) waiting to be served. The service time required by each costumer is known in advance: it is t_i minutes for costumer i. So if, for example, the costumers are served in order of increasing i, then costumer i has to wait $T_i = \sum_{j=0}^{i} t_j$ minutes. Our goal is to minimize the total waiting time

$$T = \sum_{i=0}^{n-1} T_i$$

Give an efficient algorithm for computing the optimal order in which to process the customers. The input is the list of waiting times $t_0, \ldots, t_{n-1}$. State your algorithm, analyze its running time, and prove its correctness. Pay close attention to writing clear and convincing proof of correctness.