CSOR W4246 – Fall 2018

Homework 2 theoretical

Out: Monday, October 1, 2018
Due: 8pm, Monday, October 15, 2018

Please keep your answers clear and concise. For all algorithms **you** *suggest, you must prove correctness and give the best upper bound that you can for the running time. You should always describe your algorithm clearly in English and give pseudocode.*

You should not use any external resources *for this homework, not even your textbook. Failure to follow these instructions will have a negative impact on your performance in the exams where you will have no external aids, and in interviews. For the same reason,* **you should avoid collaborating** *with your classmates, at least for the first four problems. I encourage you to work on all the recommended exercises before you solve problem 5.*

You should write up the solutions **entirely on your own**. *Collaboration is limited to discussion of ideas only. Similarity between your solutions and solutions of your classmates or solutions posted online will result in receiving a 0 in this assignment and possibly further disciplinary actions.*

You must submit your assignment as a pdf file. *Other file formats, such as jpg, doc, c, or py, will not be graded, and will automatically receive a score of 0. If you do not type your solutions, be sure that your hand-writing is legible, your scan is high-quality and your name is clearly written on your homework.*

1. (20 points) Given an undirected graph $G = (V, E)$ and two nodes $v, w \in V$, give an $O(n+m)$ algorithm that computes *the number* of shortest v-w paths in G.

2. (20 points) In cases where there are several shortest paths between two nodes (and edges have varying lengths), the most convenient among these paths is often the one with the *fewest* edges. We define

   $$best[u] = \text{minimum number of edges in a shortest path from } s \text{ to } u$$

   Give an efficient algorithm that, on input a weighted graph $G = (V, E, w)$ with positive edge weights, and an origin node $s \in V$, computes $best[u]$ for all $u \in V$.

3. (20 points) You are going on a long trip. You start on the road at mile post 0. Along the way there are n hotels, at mile posts $a_1 < a_2 < \cdots < a_n$, where each a_i is measured from the starting point. The only places you are allowed to stop are at these hotels, but you can choose which of the hotels you stop at. Your destination is the final hotel (at distance a_n) and you must stop there.

   You'd like to travel 200 miles a day, but this may not be possible, depending on the spacing of the hotels. If you travel x miles during a day, the *penalty* for that day is $(200 - x)^2$. You want to plan your trip so as to minimize the total penalty —that is, the sum, over all travel

days, of the daily penalties. Give an efficient algorithm that determines the total penalty of the optimal sequence of hotels at which to stop.

4. (20 points) A server has n customers waiting to be served. The service time for customer i is t_i minutes. So if the customers are served in order of increasing i, the i-th customer spends $\sum_{j=1}^{i} t_j$ minutes waiting to be served.

   Given n, $\{t_1, t_2, \ldots, t_n\}$, design an efficient algorithm to compute the optimal order in which to process the customers so that the total waiting time below is minimized:

$$T = \sum_{i=1}^{n} \text{(time spent waiting by customer } i\text{)}$$

5. (30 points) A challenge that arises in databases is how to summarize data in easy-to-display formats, such as a histogram. A problem in this context is the minimal imbalance problem. The input consists of an array A containing n positive numbers and an integer k. Consider k indices $j_1, j_2, \ldots, j_k$ that partition the array into $k+1$ subarrays $A[1, j_1]$, $A[j_1 + 1, j_2]$, $\ldots$, $A[j_k + 1, n]$. The weight $w(i)$ of the ith subarray is the sum of its entries. The *imbalance* of the partition is

$$\max_i \left| w(i) - \frac{\sum_{\ell=1}^{n} A[\ell]}{k+1} \right|.$$

   That is, the imbalance is the maximum deviation of any partition from the average size.

   (a) (20 points) Give an algorithm for determining the partition with minimal imbalance given A, n and k. (This basically corresponds to finding a histogram with $k+1$ bars as close to equal as possible.)

   (b) (10 points) Explain how your algorithm would change if the imbalance was redefined to be

$$\sum_i \left| w(i) - \frac{\sum_{\ell=1}^{n} A[\ell]}{k+1} \right|.$$

RECOMMENDED exercises: *do NOT return, they will not be graded.*

1. Problem 16.1 from your textbook (pp. 446-447).

2. (20 points) Consider a long country road with houses scattered very sparsely along it. (You may picture the road as a long line segment, with an eastern endpoint and a western endpoint.) You want to place cell phone base stations at certain points along the road, so that every house is within four miles of one of the base stations.

   Give an efficient algorithm that achieves this goal, using as few base stations as possible.

3. Exercise 15.4-5 from your textbook (p. 397).

4. Consider an array A with n numbers, some of which (but not all) may be negative. We wish to find indices i and j such that

   $$\sum_{k=i}^{j} A[k]$$

   is maximized. Give an efficient algorithm for this problem.

5. Given two strings $x = x_1 x_2 \cdots x_m$ and $y = y_1 y_2 \cdots y_n$, we wish to find the length of their longest common substring, that is, the largest k for which there are indices i and j such that

   $$x_i x_{i+1} \cdots x_{i+k-1} = y_j y_{j+1} \cdots y_{j+k-1}.$$

   Give an efficient algorithm for this problem.