CSE421: Introduction to Algorithms February 21, 2022

Homework 6

Yin Tat Lee Due: Feb 23, 2022 (before class)

Unless otherwise mentioned, you always need to show your algorithm's runtime and prove that it outputs the correct answer. See Homework Guideline on Ed for more details.

1. (10 Marks) Suppose that you have access to a function mydictionary that returns true if its input is a valid English word, and false otherwise. We are given as input a sentence from which the punctuation has been stripped (for example: "dynamicprogrammingisfabulous"). Assuming calls to mydictionary take unit time, give an $O(n^2)$ time algorithm to figure out whether an input string of length n can be split into a sequence of valid words or not.

2. (10 Marks) Suppose we want to replicate a file over n servers, labeled $S_1, S_2, \ldots, S_n$. To place a copy of a file at server S_i results in a placement cost of c_i for an integer $c_i > 0$.

   Now, if a user requests the file from server S_i, and no copy of the file is present at S_i, then the servers $S_{i+1}, S_{i+2}, \ldots, S_j$ are searched, where $j > i$ and S_j is the next server holding a copy. This results in an access cost of $j - i$. The access cost is 0 if S_i holds a copy of the file. We will require that a copy of the file be placed at server S_n so that all such searches will terminate, at the latest at S_n.

   We know that a_i many users will try to first access the file at server S_i. Find a polynomial time algorithm that computes on which subset of servers a copy of the file should be placed so that the sum of all the placement cost plus all the access cost is minimized.

3. (10 Marks) Given a rows $\times$ cols binary matrix filled with 0's and 1's, give a polynomial time algorithm to return the largest rectangle containing only 1's and return its area. Such tasks routinely appear in image processing in various contexts.

   As an example,

   Input: $\mathsf{matrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$

   Output: 6

   Please note that the better your runtime is, the higher your score.

4. (**Extra Credit)** Given a directed graph G with n vertices (assume all edges have unit length). Our goal is to compute the shortest path distance from i to j for all pairs of vertices.

   - Show how to find all pair shortest path distances in $O(n^3)$ time using dynamic programming.
   - Google Seidel's algorithm and explain how to find all pair shortest path distances in $O(n^{\log_2 7} \log n)$ time in your own word.