CSE421: Introduction to Algorithms January 26, 2022

Homework 4

Yin Tat Lee Due: Feb 2, 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) Solve the following recurrences. Some questions cannot be solved by the master theorem. (No proof is required)

   (1) $T(n) = T(n/2) + 2$

   (2) $T(n) = 2T(n/2) + n$

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

   (4) $T(n) = 7T(n/2) + n^2$

   (5) $T(n) = 8T(n/2) + n^2 \log^3 n$

2. (10 Marks) Suppose you are given a sequence of n objects $x_1, x_2, \ldots, x_n$, but you only have access to them through an oracle that, given i and j, can determine if $x_i = x_j$ or not. Design an algorithm that outputs whether there exists an index i such that there are *strictly* greater than $\frac{n}{3}$ objects in the sequence $x_1, \ldots, x_n$ that are equal to x_i. Your algorithm must run in time $O(n \log n)$ (and make at most $O(n \log n)$ calls to the oracle).

3. (10 Marks) Let $G = (V, E)$ be an undirected graph and let $w(e) \geq 0$ for all edge weights for $e \in E$. Recall that a subset $M \subseteq E$ is called a *matching* if no two edges in M share a node. The *Maximum Weight Matching* problem consists of finding a matching M so that the weight $w(M) := \sum_{e \in M} w(e)$ is maximized. This is indeed a polynomial time solvable problem, but the algorithm is highly non-trivial and way beyond the scope of this course. However, it is easy to design a greedy algorithm that finds a 2-approximation. To be precise we suggest the following algorithm:

   1: Sort the edges so that $w(e_1) \geq w(e_2) \geq \ldots \geq w(e_m)$
   2: Set $M := \emptyset$
   3: **for** i from 1 to m **do**
   4: If $M \cup \{e_i\}$ is still a matching then update $M := M \cup \{e_i\}$

   Let us denote $\texttt{OPT} := \max\{w(M) : M \subseteq E \text{ is a matching}\}$ as the value of the optimum solution and `GREEDY` as the value of the solution produced by the greedy algorithm above. Solve the following:

   (a) Show that for any $\varepsilon > 0$ there is an instance where $\texttt{GREEDY} \leq (\frac{1}{2} + \varepsilon) \cdot \texttt{OPT}$.

   (b) Suppose that $\{a_1, \ldots, a_p\} \subseteq E$ are the edges selected by greedy and suppose that $\{b_1, \ldots, b_q\} \subseteq E$ are the edges selected by the optimum solution. Moreover suppose those edges are sorted so that $w(a_1) \geq w(a_2) \geq \ldots \geq w(a_p)$ and $w(b_1) \geq w(b_2) \geq \ldots \geq w(b_q)$. For the sake of simplicity, assume that q is even. Show that $w(a_i) \geq w(b_{2i-1})$ for all $i \in \{1, \ldots, \frac{q}{2}\}$.

(c) Prove that $\texttt{GREEDY} \geq \frac{1}{2} \cdot \texttt{OPT}$.

4. (**Extra Credit)** You are Elizabeth Holmes. Your company Theranos focuses on blood testing. Suppose you need to analyze N people's blood for DIVOC-91. For simplicity, suppose the followings:

   - There are exactly d patients with DIVOC-91 where $d \ll N$
   - You can mix the blood samples of multiple patients and put it in the machine to run the analysis. If any patient in the mixture has DIVOC-91, it outputs positive (otherwise, negative).
   - You have many such machines, but each run takes 1 day and 1 million dollars.

   Show how to find the d patients with DIVOC-91 using these machines and at most $O(d \log N)$ dollars and $O(\log N)$ days. Hint: This is a divide and conquer question.

   (For a real challenge, design an algorithm that uses $O(d \log N)$ dollars and 1 day. You probably need randomization techniques for the 1 day algorithm.)