Assignment 1

COMP 3804, Fall 2022

Upload in Brighspace by 11:59 PM on October 6, 2022

1 Guidelines

General guidelines are as follows:

1. Since we are only accepting assignments via the Brighspace system, and we will discuss the solutions immediately after the due date, no late submissions will be entertained after the cut-off time & date.

2. Please write clearly and answer questions precisely. It is your responsibility to ensure that what is uploaded is readable. If we can't read, we can't mark!

3. Please cite all the references (including web-sites, names of friends, etc.) that you used/consulted as the source of information for each of the questions.

4. The first ten questions are worth 10 points each, totalling 100. The maximum mark for this assignment is 100. Bonus problem has ten extra points.

5. When a question asks you to design an algorithm - it **requires** you to

   (a) Clearly spell out the **steps** of your algorithm as a pseudo code.

   (b) **Prove** that your algorithm is correct

   (c) **Analyze** the running time.

6. This assignment is mostly based on the material that should have been learnt in COMP 1805/2402/2804. The purpose is to ensure that you have a good grasp of the background material to understand various topics in this course. If you have difficulty answering any of these questions, it is your responsibility to review them quickly.

2 Problems

1. (a) Define the functions $O()$, $\Omega()$ and $\Theta()$ that occur frequently in analyzing the complexity of algorithms (refer to your notes/textbooks of COMP 1805/2402/2804).

   (b) Let $p(n) = a_d n^d + a_{d-1} n^{d-1} + \cdots a_1 n + a_0$, where $a_d > 0$, be a d-degree polynomial in n. Also $a_0, \cdots, a_d$ are positive constants. Let k be a positive integer. Using your definitions in Part (a) show that:

      i. If $k > d$, then $p(n) \in O(n^k)$.

      ii. If $k < d$, then $p(n) \in \Omega(n^k)$.

      iii. Is $20n^3 + 10n^2 - 100n - 5 \in O(n^4)$?

      iv. Is $20n^3 + 10n^2 - 100n - 5 \in \Omega(n^2)$?

2. Evaluate the following recurrences (You can assume that $T(1) = 1$ for each of them. Don't worry about ceilings and floors.)

   (a) $T(n) = 5T(n/5) + O(n)$

   (b) $T(n) = T(8n/9) + O(n^2)$

   (c) $T(n) = T(n-2) + O(1)$

   (d) $T(n) = \sqrt{n}T(\sqrt{n}) + n$

3. Suppose you need to choose between the following algorithms that solve the same problem:

   (a) Algorithm A solves the problem by dividing it into 4 subproblems of half of the size, recursively solves each of them, and combines the solution in linear time.

   (b) Algorithm B solves the problem of size n by recursively solving three subproblems of size $n - 6$ and then combining the solutions in constant time.

   (c) Algorithm C solves the problem of size n by dividing it into 9 subproblems of size $n/3$ each, recursively solving each of them, and then combining the solution in $O(n^2)$ time.

   What are the running times of each of these algorithms? Which one will you choose and Why?

4. Recall that the analysis of finding the k-th smallest element in a set S of n-distinct elements led us to the recurrence for the time complexity as $T(n) \leq T(\frac{n}{5}) + T(\frac{7}{10}n) + O(n)$. We showed, by using induction on n, that $T(n) = O(n)$. Show that the recurrence $T(n) \leq T(\alpha n) + T(\beta n) + O(n)$ evaluates to $O(n)$ for any choice of positive reals α and β, where $\alpha + \beta < 1$. We can assume that for small values of n, $T(n) = O(1)$.

5. You are given an array A consisting of n positive integers, where each element is $\leq 1000n$. Devise an algorithm, running in $O(n)$ time, to sort A in ascending order. Justify your answer.

6. Recall (from your COMP 2402 course) that there is a lower bound for sorting. Answer the following questions

   (a) What does it mean to have a lower bound for a problem?

   (b) State clearly what is the lower bound claim for sorting a set of n (real)-numbers.

   (c) Why this claim does not apply to the problem in Question 5?

7. You are given an array A consisting of n real numbers. Describe and analyze an algorithm, running in $O(n)$ time, that rearranges the elements of A so that A forms a binary heap. Once A is transformed into a Binary Heap, show how you can report the elements in A in sorted (ascending) order. How much time it takes to report all the elements of A in the sorted order? Justify your answer.

8. You are given a set of n real numbers which you are asked to insert incrementally in an initially empty binary search tree. Note that the time to insert an element in a binary search tree of size x is $O(\log x)$. What is the total running time of inserting all the n elements in the tree. Justify your answer. Assume that you have formed the binary search tree on n elements, show how you can report the elements in a sorted order in $O(n)$ time.

9. Reflecting on the answers to the previous two questions, is the construction of a binary heap in $O(n)$ time or reporting the elements in sorted order from a binary search tree in $O(n)$ time is in contradiction to the lower bound for sorting (refer to Question 6)? Justify your answer.

10. We say two $n \times n$ matrices A and B are equal, denoted by $A = B$, if $A_{ij} = B_{ij}$ for all $1 \leq i, j \leq n$. We say a matrix $A = 0$ if $A_{ij} = 0$ for all $1 \leq i, j \leq n$. We say $A \neq 0$ if there exists some i, j for which $A_{ij} \neq 0$.

    Suppose you are given three $n \times n$ matrices X, Y, and Z. You need to determine $Z \stackrel{?}{=} XY$. Answer the following

    (a) Can we determine $Z \stackrel{?}{=} XY$ in time proportional to multiplying two $n \times n$ matrices?

    (b) Let A be a matrix of dimensions $n \times n$. Let $v = (v_1, \ldots, v_n)$ be a random Boolean vector of length n, i.e., each of its coordinate v_i is set either to 0 or 1 independently and with equal probability. Consider Av, i.e. the product of the matrix A with the vector v and it results in a vector of length n. Show that for $A \neq 0$, $Pr(Av = 0) \leq \frac{1}{2}$.

    (c) Assume $XY \neq Z$. Show that for any random Boolean vector v of length n, $Pr(XYv = Zv) \leq \frac{1}{2}$.

    (d) Show that the product XYv can be computed in $O(n^2)$ time.

    (e) Show that we can determine $XYv \stackrel{?}{=} Zv$ in $O(n^2)$ time.

(f) Conclude that we can design a randomized test for checking $XY \stackrel{?}{=} Z$ that runs in $O(n^2)$ time. (Thus, it is more efficient to verify $Z \stackrel{?}{=} XY$ than multiplying matrices.)

11. (Bonus Problem:) Assume that you have a set of three parallel and distinct lines in plane. Assume that Line 1 contains a set R of n red points, Line 2 contains a set B of n blue points, and Line 3 contains a set G of n green points, where n is a positive integer. You need to design an algorithm that determines if there is a triplet of points (r, b, g) such that they lie on a line segment, where $r \in R$, $b \in B$ and $g \in G$. (Observe that it is straightforward to design an algorithm running in $O(n^3)$ time by considering each choice for points r, b, and g, and in $O(1)$ time testing whether they all lie on a segment.) To get Bonus points, design an algorithm running in $O(n^2)$ time.