CECS 528, Exam 2 Version b, March 24th, Spring 2023, Dr. Ebert

Rules for Completing the Problems

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION allowed when solving these problems. Make sure all these items are put away BEFORE looking at the problems. FAILURE TO ABIDE BY THESE RULES MAY RESULT IN A FINAL COURSE GRADE OF F.

Directions

Choose up to **six problems** to solve. Clearly mark each problem you want graded by placing an 'X' or check mark in the appropriate box. **If you don't mark any problems or mark more than six, then we will record grades for the six attempted problems that received the *fewest* points.**

Problem	1	2	3	4	5	6	LO1	LO2	LO3	LO4
Grade?										
Result										

Your Full Name:

Class ID Number:

1. Note: correctly solving part a of this problem counts for passing LO5.

   a. Demonstrate the partitioning step of Hoare's version of `Quicksort` for the array

   $$a = 11, 6, 3, 10, 9, 5, 2, 8, 1, 7, 4,$$

   where we assume that the pivot equals the median of the first, last, and middle integers of a. Provide arrays a_{left} and a_{right}. (10 pts)

   b. What is the worst-case running time of Hoare's algorithm in the case that the members of the input array a form a strictly decreasing sequence of integers. Defend your answer. Again, we assume that the pivot equals the median of the first, last, and middle integers of a. (15 pts)

2. Given that $r = ae + bg$, $s = af + bh$, $t = ce + dg$, and $u = cf + dh$ are the four entries of AB, and Strassen's products are obtained from matrices

$$A_1 = a, B_1 = f - h, A_2 = a + b, B_2 = h, A_3 = c + d, B_3 = e, A_4 = d, B_4 = g - e,$$

$$A_5 = a + d, B_5 = e + h, A_6 = b - d, B_6 = g + h, A_7 = a - c, B_7 = e + f,$$

Compute $P_1, \ldots, P_7$ and use them to compute r, s, t, and u. Note: correctly solving this problem counts for passing LO6. (25 pts)

3. Answer the following. Note: correctly solving this problem counts for passing LO7.

   a. Provide the dynamic-programming recurrence that is used in the Floyd-Warshall algorithm. (10 pts)

   b. When executing the Floyd-Warshall algorithm, assume

$$d^4 = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 & 7 \\ 18 & 0 & 3 & 19 & 6 & 14 \\ 16 & 4 & 0 & 6 & 5 & 1 \\ 15 & 10 & 19 & 0 & 6 & 17 \\ 4 & 1 & 1 & 3 & 0 & 1 \\ 4 & 6 & 1 & 3 & 5 & 0 \end{pmatrix}$$

   has been computed. Use this matrix to compute d^5. Then use d^5 to compute d^6. (15 pts)

4. Given an array of n integers, a **local maximum** for a is a value of a occurring at some index i, $0 < i < n-1$, for which $a[i] > a[i-1]$ and $a[i] > a[i+1]$. Describe an optimal (with respect to big-O running-time) algorithm for determining whether or not a has a local maximum. Please do *not* use pseudocode. Rather, number each step of your algorithm and clearly explain each step. Before beginning the first step clearly define all variables used by the algorithm and indicate their initial values. Provide the running time of your algorithm and justify your answer. Suboptimal and/or poorly written algorithms will be awarded 0 points. (25 pts)

5. An instance of the `k Nearest Neighbor` problem is a point $\vec{y} \in R^n$ and points $\vec{x}_1, \ldots, \vec{x}_n \in R^n$, and the problem is to determine the $k > 0$ points in $\vec{x}_1, \ldots, \vec{x}_n$ that are nearest to $\vec{y}$ in terms of Euclidean distance $d(\vec{y}, \vec{x}_i)$, where $0 < k \leq n$. Describe an $O(n)$ algorithm for determining $\vec{y}$'s k nearest neighbors. Please do *not* use pseudocode. Rather, number each step of your algorithm and clearly explain each step. Before beginning the first step clearly define all variables used by the algorithm and indicate their initial values. Provide the running time of your algorithm and justify your answer. Suboptimal and/or incompletely/vaguely described algorithms will be awarded 0 points. Hint: $O(kn)$ grows faster than $O(n)$. (25 pts)

6. Consider the problem of counting the number of times a bit string u appears in a bit string v, where we assume u's bits do *not* have to appear consecutively. For example, if $u = 011$ and $v = 001011$, then u appears seven times in v at the following index locations:

$$(135), (136), (156), (235), (236), (256), \text{ and } (456).$$

Let $N(i, j)$ denote the number of times u-prefix $u_1 \cdots u_i$ appears in v-prefix $v_1 \cdots v_j$.

a. Provide a dynamic-programming recurrence for $N(i, j)$. Hint: an appearance of the u-prefix in the v-prefix may or may not make use of bit j of the v-prefix. (15 pts)

b. Apply your recurrence to the problem instance $u = 101$ and $v = 1101011$. Provide the matrix of subproblem solutions. (10 pts)

LO1. Suppose $f(n)$ is a function that has sublinear growth. Prove that $n^{f(n)}$ does *not* have exponential growth.

LO2. For the weighted graph with edges

$$(a, c, 5), (b, c, 4), (c, e, 6), (c, f, 3), (c, d, 2), (d, f, 1),$$

Show how the forest of M-Trees changes when processing each edge in Kruskal's sorted list of edges. When unioning two trees, use the convention that the root of the union is the root which has the *lower* alphabetical order. For example, if two trees, one with root a, the other with root b, are to be unioned, then the unioned tree should have root a.

LO3. In the correctness proof of Prim's algorithm, suppose $T = e_1, \ldots, e_{n-1}$ are the edges selected by Prim's algorithm (in that order) and T_{opt} is an mst that uses edges $e_1, \ldots, e_{k-1}$, but for which $e_k \notin T_{\text{opt}}$. Explain how to identify an edge $e \in T_{\text{opt}}$ for which $T_{\text{opt}} + e_k - e$ is an mst that now possesses edges $e_1, \ldots, e_k$. Hint: consider tree T_{k-1} which is Prim's tree after round $k-1$ and is contained in T_{opt}.

LO4. Given $T_1(n) = 64T_1(n/4) + n^3$, and $T_2(n) = aT_2(n/3) + n^2$ what is the greatest possible value that a can assume, and still have $T_2(n) = o(T_1(n))$? Show work and explain.