Fall 2024: CSE 317 : Introduction to Algorithms

Sample Questions

Question 1 [10 points]

Let $X = \langle x_1, x_2, \ldots, x_n \rangle$ be a sequence of n integers. We say the sequence $S = \langle s_1, s_2, \ldots, s_n \rangle$ is the *sequence of partial sums the sequence* X, if:

$$\begin{aligned}
s_1 &= x_1 \\
s_2 &= x_1 + x_2 = s_1 + x_2 \\
s_3 &= x_1 + x_2 + x_3 = s_2 + x_3 \\
&\vdots \\
s_n &= x_1 + x_2 + \cdots + x_n = s_{n-1} + x_n
\end{aligned}$$

Design a *parallel algorithm* with n processors to compute partial sums.

Question 2 [5 points]

Design a *sequence algorithm* for the problem from previous question to computer partial sums. Also, provide the worst-case running time of your algorithm.

Question 3 [50 points]

Prove following problems to **NP**-complete. [Note: These are *really difficult problems.*]

(a) **Name:** Graph 3-Colorability 3COL
Input: An n-node undirected graph $G(V, E)$ with node set V and edge set E.
Question: Can each node of $G(V, E)$ be assigned exactly one of three colors – `Red`, `Blue`, `Green` – in such a way that no two nodes which are joined by an edge, are assigned the same color?

(b) **Name:** Monochromatic triangle MONO TRIANGLE
Input: An n-node undirected graph $G(V, E)$ with node set V and edge set E.
Question: Can the edges, E, of G be partitioned into two disjoint sets E_1 and E_2, in such a way that neither of the two graphs $G_1(V, E_1)$ or $G_2(V, E_2)$ contains a triangle, i.e. a set of three distinct nodes u, v, w such that $\{u, v\}$, $\{u, w\}$, $\{v, w\}$ are all edges?

(c) **Name:** Multiprocessor Scheduling MULTIPROCESSOR SCHEDULING
Input: A set, T, of tasks and for each task t in T a positive (integer) running time $len(t)$. A positive integer D, called the deadline.
Question: Is there a 2-processor schedule for the tasks that completes within the deadline, D, i.e. is there a function $\sigma : T \to \mathbb{N}$ such that both of the following hold:

- For all $u \geq 0$ the number of tasks t in T for which $\sigma(t) \leq u < \sigma(t) + len(t)$ is at most 2.
- For all tasks t, $\sigma(t) + len(t) \leq 0$

(d) **Name:** Maximum 2-Satisfiability MAX 2 SAT
Input: A set of m clauses $C_1, C_2, \ldots, C_m$ over n Boolean valued variables X_n, where each clause depends on two distinct variables from X_n; a positive integer k with $k \leq m$.
Question: Is there an assignment of Boolean values to the variables X_n such that at least k distinct clauses take the value true under the assignment?

(e) Name: Shortest Common Superstring SHORTEST COMMON SUPERSTRING
Input: A finite set $R = \{r_1, r_2, \ldots, r_m\}$ of binary strings (sequences of 0 and 1); positive integer k.
Question: Is there a binary string w of length at most k such that every string in R is a substring of w, i.e. for each r in R, w can be decomposed as $w = w_0 r w_1$ where w_0, w_1 are (possibly empty) binary strings?

Question 4 [10 points]

Given an array of n elements containing all but one of the integers from 1 to $n+1$. Design an $O(n)$ algorithm to find the missing integer. Assume that the input array is *not sorted*. Also, provide the worst-case running time of your algorithm.

Question 5 [10 points]

Divide and conquer is a technique where the input problem is decomposed into subproblems of smaller sizes recursively and solutions of these subproblems are used to construct a solution to the main input problem.

(a) (5 points) Explain the difference between dynamic programming and divide and conquer technique.

(b) (5 points) The Fibonacci sequence is a famous mathematical sequence. The first and second Fibonacci numbers are 1 and 1. The n-th Fibonacci number is defined as the sum of the previous two Fibonacci numbers, where $n > 2$. Design an $O(n)$ dynamic programming algorithm to compute the n-th Fibonacci number.

Question 6 [10 points]

Given an array of distinct integers $A = \langle a_1, \ldots, a_n \rangle$ where $n > 0$. A pair of elements (a_i, a_j) where $i < j$ and $a_i > a_j$ is called an *inversion.*

(a) Design an $O(n \log n)$ *divide and conquer* algorithm that counts the number of inversions in A.

(b) Analyze the time complexity of your algorithm using the *Master Theorem.*

Question 7 [10 points]

IBA has a wonderful architecture consisting of many interconnected buildings and pathways (streets). Suppose the n buildings at IBA are labeled as $1, ..., n$ and $n-1$ two-way streets. A street $s_i = (a_i, b_i)$ connects buildings a_i and b_i where $1 \leq a_i < b_i \leq n$ and $1 \leq i \leq n-1$. We know that starting from any building we can reach any other building walking through the streets. Setting up a lamp in front of a building lights all the streets adjacent to it. Design an $O(n)$ *dynamic programing* algorithm to find the minimum number of lamps required to light all the streets at IBA.

Question 8 [10 points]

Consider five matrices of following respective sizes:

$$M_1 : 3 \times 4, \quad M_2 : 4 \times 8, \quad M_3 : 8 \times 2, \quad M_4 : 2 \times 9.$$

Use *dynamic programing* to find the minimum number of scalar multiplications required to compute the product $\prod_{j=1}^{4} M_j$ as well as an optimal *parenthesization.*

Question 9 [10 points]

Given a directed graph $G = (V, E)$ with n vertices and m edges. Design an $O(n + m)$ algorithm to find the number of strongly connected components in G. Also, provide the worst-case running time of your algorithm.

Question 10 [10 points]

Let $A = [a_1, a_2, \ldots, a_n]$ be an array of n integers and x be an integer. Design an efficient algorithm to find if for any two elements a_i and a_j in A such that $|a_i - a_j| = x$. Also, provide the worst-case running time of your algorithm.

Question 11 [10 points]

Let $S = \{x_1, x_2, \ldots, x_n\}$ be a set of n positive integers. Design a randomized algorithm with probability $1 - 1/n$ that finds an element $x \in S$ such that x is the upper half when S is sorted.

Question 12 [10 points]

Let $A = \langle a_1, a_2, \ldots, a_n \rangle$ be a sequence of n elements that contains exactly k occurrences of the element x, here $1 \leq k \leq n$. Design a randomized algorithm to find an index j such that $a_j = x$. Compute the expected running time of your algorithm.

Question 13 [10 points]

Let $A_1, A_2, \ldots, A_n$ be n arrays of integers where each array A_i is sorted in non-decreasing order and has corresponding size n_i. Design an efficient greedy algorithm to merge all the arrays into a single sorted array. Also, provide the worst-case running time of your algorithm.

Question 14 [10 points]

Let $G = (V, E)$ be an undirected graph such that for every edge $e \in E$, $c_e > 0$ is the cost of the edge e.

Prove/disprove following statements:

(a) Let e be an edge with the minimum cost in G then every minimum spanning tree of G contains the edge e.

(b) Let e be an edge with maximum cost in G then no minimum spanning tree of G contains the edge e.

(c) Let all the costs of edges be unique then there is a unique minimum spanning tree of G (regardless of starting vertex).

(d) Let all the costs of edge be the same then there are multiple minimum spanning trees of G.

(e) Let v be a vertex in G with degree 1 then v is not included in any minimum spanning tree of G.

(f) Let $e = (u, v)$ be an edge with the maximum cost in G and degree of v is 1 then e is included in every minimum spanning tree of G.