CSE 317: Design and Analysis of Algorithms

Sample Questions: Final

1. Consider the following algorithm DIVIDE:

   Algorithm: DIVIDE

   Input: $y, z \in \mathbb{N}$

   Output: The product $q, r \in \mathbb{N}$ s.t., $y = qz + r$

   $r \leftarrow y$, $q \leftarrow 0$, $w \leftarrow z$
   while $w \leq y$
       $w \leftarrow 2w$
   while $w > z$
       $q \leftarrow 2q$, $w \leftarrow \lfloor w/2 \rfloor$
       if $w \leq r$ **then**
           $r \leftarrow r - w$, $q \leftarrow q + 1$
   return (q, r)

   (a) Trace the steps of the algorithm for: DIVIDE$(10, 3)$.

   (b) Prove that this algorithm is correct.

   (c) Express the time complexity of this algorithm in O-notations.

2. Let $A[1 \ldots n]$ be an array of n elements. An element x is called a *majority element* if it occurs at least $\lceil n/2 \rceil$ times in A. Assume that a majority element is guaranteed to exist.

   (a) Design a *Monte Carlo randomized algorithm* that outputs a majority element of A.

   (b) Prove that your algorithm outputs the correct answer with probability at least $1 - 1/n$.

   (c) Analyze the expected running time of your algorithm.

   You may assume that equality comparisons between array elements take constant time.

3. Show following problems are in NP.

   (a) **Dominating set problem:** Given an undirected graph $G = (V, E)$, a subset of vertices $D \subseteq V$ is called a *dominating set* if for every vertex $u \in V \setminus D$, there is a vertex $v \in D$ such that $\{u, v\} \in E$.

   Question: Given an undirected graph G and an integer k, does G contain a dominating set of size at most k?

   (b) **Hamiltonian path problem:** Given an undirected graph $G = (V, E)$, a path in G that visits each vertex $v \in V$ exactly once is called *Hamiltonian path*.

   Question: Given an undirected graph G, does G contain a Hamiltonian path?

4. Let A be an array of n elements such that only $\log_2 n$ elements in A are unique. For example, if $A = \langle 9, 0, 0, 9, 1, 0, 1, 9 \rangle$ then $|A| = 8$ however only $\log_2 8 = 3$ elements in A are unique i.e., $0, 1, 9$.

   Solve following questions to sort A in ascending order.

   (a) Design an $O(n \log n)$ algorithm to sort A.

   (b) Design an $O(n \log \log n)$ algorithm to sort A.

   (c) Design an $O(n)$ algorithm to sort A.

5. (10 points) You are given an integer array nums $= \langle a_1, a_2, \ldots, a_n \rangle$ of n non-negative integers, where each element a_i represents the *maximum jump length* from position i. You start at the first index of the array, and your goal is to reach the last index in the minimum possible number of jumps.

   Design a *dynamic programming* algorithm to compute the minimum number of jumps required to reach the last index. If it is not possible to reach the last index, your algorithm should return -1.

   Example 1:

   Input: $\langle 2, 3, 1, 1, 4 \rangle$, Output: 2

   Explanation: Jump one step from index 1 to index 2, then three steps to the last index.

   Example 2:

   Input: $\langle 3, 2, 1, 0, 4 \rangle$, Output: -1

   Explanation: You will always get stuck at index 4, whose maximum jump length is 0, making it impossible to reach the last index.

6. Given a sequence of numbers $a_1, a_2, \ldots, a_n$, $n \geq 3$ of distinct integers, a triplet a_i, a_j, a_k is called a *strange triplet* if $a_i > a_j > a_k$ whenever $i < j < k$. Compute the average number of strange triplets in any given sequence of unique numbers uniformly drawn from the set of integers.

7. Consider following randomized strategy to check if the given input number q is prime?

   Algorithm: PRIME

   Input: A positive integer $N \geq 2$

   Output: true if N is prime, false otherwise with some error probability p

   1. $a \leftarrow$ RANDOM-PRIME($\sqrt{N}$) // *a is random prime betwen 2 and* $\sqrt{N}$
   2. $b \leftarrow$ RANDOM-PRIME($\sqrt{N}$) // *b is random prime betwen 2 and* $\sqrt{N}$
   3. **if** $ab = N$ **then return** true
   4. **else return** false

   Find the probability of error in the above algorithm.

8. Let $G = (V, E)$ be a directed acyclic graph and s and t be two vertices in G. Design a dynamic programming algorithm compute the number of all directed paths between s and t.

9. Let $G = (V, E)$ be a directed acyclic graph, a vertex $s \in V$ is called a *universal sink* without any outgoing edges such that every other vertex $u \in V$ has a directed path to v. Design an efficient algorithm find a universal sink vertex in any given DAG.

10. IBA is planning to build a new campus with n buildings labeled as $1, \ldots, n$. There will be $n - 1$ two-way streets in this new campus.

    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 campus streets. Setting up a lamp in front of a building lights all the streets adjacent to it. Design an $O(n)$ dynamic programming algorithm to find the minimum number of lamps required to light all the campus streets.