CIT 596 Online **Spring 2020**

Module 5–6

Name: NAME HERE

Questions requiring written answers.

1. You are given an array $A[1..n]$ of n distinct elements such that each element in A is at most k positions away from its position in sorted order ($2 \leq k < n$). Give an $O(n \log k)$ time algorithm to sort the array. Prove your algorithm correct and analyze its running time.

2. You are given an array $A[1..n]$ of n integers. We say the array is *well-positioned* with respect to a parameter $k \in [1..n]$ iff for any k consecutive indices in the array, there are no two indices such that the value of one index is at least twice the value as the other. Give an $O(n \log k)$ time algorithm to determine if A is well-positioned. Prove your algorithm correct and analyze its running time.

3. You are given three arrays of integers A, B, and C with n elements each as well as a target integer t. Design an algorithm to determine if there exists three numbers $a \in A, b \in B$, and $c \in C$ such that $a + b + c = t$. Prove your algorithm correct and analyze its running time. Your algorithm should run in worst-case $O(n^2 logn)$ or $O(n^2)$ expected time.

4. You are given a directed graph $G = (V, E)$ with n vertices labeled 1 through n. For each vertex u, let $H(u)$ be the highest-valued vertex that is reachable from u. Design an efficient algorithm to compute $H(u)$ for each vertex in the graph.

5. Given a *tree* there is a unique simple path between any two vertices. Give an efficient algorithm to find the length of the longest such path.

6. You are given a directed graph $G = (V, E)$. Let S be the subset of vertices in G that are able to reach some cycle. Design an efficient algorithm to compute S.

7. You are given a directed acyclic graph $G = (V, E)$, where each vertex v that has indegree 0 has a value $value(v)$ associated with it. For each other vertex u, define $Pred(u)$ to be the set of vertices that have incoming edges to u. We now define $value(u) = \sum_{v \in Pred(u)} value(v)$. Design an efficient algorithm to compute $value(u)$ for all vertices u. Analyze the running time of your algorithm and prove it correct