CSE 431/531: Algorithm Analysis and Design Fall 2021

Homework 1

Instructor: Shi Li **Deadline: 9/28/2021**

Your Name: _____ Your Student ID: _____

Problems	1	2	3	Total
Max. Score	20	30	30	80
Your Score				

Problem 1. For each pair of functions f and g in the following table, indicate whether $f = O(g)$, $f = \Omega(g)$ and $f = \Theta(g)$ respectively. Then prove $\lceil 10n^{1.9} \rceil = O(n^2)$.

$f(n)$	$g(n)$	O	Ω	Θ
$n^2 - 3n + 10$	n			
$\log_3 n$	$10 \log_2(n^3)$			
$10n^2 - n$	$n^2 \log n$			

Problem 2. Given a *sorted* array A of size n, design an algorithm to check if there are two numbers in the array whose sum is 0. That is, decide whether there are two indices $i, j \in \{1, 2, 3 \cdots, n\}$ such that $A[i] + A[j] = 0$. (The two indices can be the same; thus if the array contains the number 0, we should output "yes".)

Example: if the input is $(-8, -5, -2, 1, 4, 6, 8, 9)$, then the output is "yes" since $(-8)+8 = 0$. If the input is $(-8, -5, -2, 1, 4, 6, 7, 9)$, then the output is "no". Design an $O(n)$-time algorithm for the problem.

Problem 3. A cycle in a *directed* graph $G = (V, E)$ is a sequence of $t \geq 2$ *different* vertices $v_1, v_2, \cdots, v_t$ such that $(v_i, v_{i+1}) \in E$ for every $i = 1, 2, \cdots, t-1$ and $(v_t, v_1) \in E$. See Figure 1 for an example. Given the linked-list representation of a directed graph $G = (V, E)$ with $n = |V|$ and $m = |E|$, design an $O(n+m)$-time algorithm to decide if G contains a cycle or not; if it contains a cycle, output one (you only need to output one cycle).

You can use topological ordering as a sub-procedure. If you do, it is not necessary to write down the whole pseudo-code for the procedure.

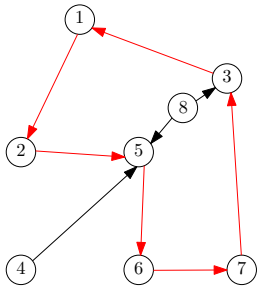

Figure 1: A cycle (denoted by red edges) in a directed graph.