NAME: ______________________________

CSE 421
Introduction to Algorithms
Midterm Exam Spring 2021

DIRECTIONS:

- Answer the problems on the exam paper.
- Justify all answers with proofs (except for Problem 1), unless the facts you need have been stated or proven in class, or in Homework, or in sample-midterm.
- If you need extra space use the back of a page or two additional pages at the end
- You have 70 minutes to complete the exam.
- Please do not turn the exam over until you are instructed to do so.
- Good Luck!

1	/30
2	/20
3	/20
4	/20
Total	/90

1. (25 points, 5 each) For each of the following problems answer **True** or **False** (no proof needed).

   (a) $n + \log n = \Omega(n \log n)$.

   (b) Every (not necessarily connected) graph with n edges has exactly one cycle.

   (c) In every DAG with n vertices, for any $1 \leq k \leq n-1$, there are at most k vertices with out-degree at least $n-k$.

   (d) A graph G has exactly *three* connected components if and only if there are exactly *two* cuts $(S_1, V - S_1), (S_2, V - S_2)$ of G with *no* edges in them (i.e., every other cut has at least one edge).

   (e) The Kruskal's algorithm runs in time $\Theta(m \log m)$.

   (f) If $T(n) \leq 27T(n/9) + n^3$, $T(1) = 1$, then $T(n) = O(n^3 \log n)$.

2. Given a connected undirected weighted graph $G = (V, E)$ where every edge $e \in E$ has a *positive integer* weight w_e such that the sum of weights of all edges is at most $4m$, i.e., $\sum_{e \in E} w_e \leq 4m$, and a vertex $s \in V$, design an $O(m + n)$ time algorithm that outputs the length of the shortest path from s to all vertices of V. Recall that in a weighted graph the length of a path P with edges $e_1, \ldots, e_k$ is $w_{e_1} + \cdots + w_{e_k}$.

3. Given *sorted* array of n *distinct even* integers, arranged in *increasing* order $A[1, n]$, you want to find out whether there is an index i for which $A[i] = 2i$. Give an algorithm that runs in time $O(\log n)$ and outputs "yes" if such an i exists and "no" otherwise. (Recall that an integer is even if it is a multiple of 2).

4. Show that there are at least $3 \cdot 2^{n-1}$ ways to properly color vertices of a tree T with n vertices using 3 colors, i.e., to color vertices with three colors such that any two adjacent vertices have distinct colors. Note that it can be shown that there are exactly $3 \cdot 2^{n-1}$ ways to properly color vertices of T with 3 colors but in this problem, to receive full credit, it is enough prove the "at least" part.

   For example, there are (at least) $3 \cdot 2^2 = 12$ ways to color a tree with 3 vertices as show below: