NAME: ______________________________

# CSE 421
# Introduction to Algorithms
# Midterm Exam Spring 2018

Lecturer: Yin Tat Lee — 30 April 2018

DIRECTIONS:

- Answer the problems on the exam paper.
- You may use facts proven in class.
- If you need extra space use the back of a page.
- You have 50 minutes to complete the exam.
- Please do not turn the exam over until you are instructed to do so.
- Good Luck!

| | |
|---|---|
| 1 | /40 |
| 2 | /20 |
| 3 | /20 |
| 3 | /20 |
| Total | /100 |

**Problem 1 (40 points, 4 each).**
For each of the following problems circle **True** or **False**. **NO** justification is needed.

1. $n^{1.2} = O(n2^{\sqrt{\log n}})$. True / False

2. $n^{10} = O(2^{n^{0.1}})$. True / False

3. Any graph $G$ with no cycles has exactly $n-1$ edges. True / False

4. Dijkstra's algorithm works for directed graph with positive length. True / False

5. In the Union Find data structure, every tree representing a connected component can have depth at most $O(1)$. True / False

6. For the stable matching problem, there can be more than 1 solution. True / False

7. There is a polynomial time algorithm for testing if an undirected graph is 2-colorable or not. . True / False

8. Let $T$ be a breadth-first search tree of a undirected graph. Let $(x, y)$ be an edge of $G$ that is not an edge of $T$, then one of $x$ or $y$ is an ancestor of the other. True / False

9. If $T(n) \leq 2T(n/2) + n$, then $T(n) = O(n \log^2 n)$. True / False

10. Let $C$ be any cycle in a graph $G$ with distinct costs, and let edge $e$ be the cheapest edge belonging to $C$. Then $e$ belongs to all minimum spanning tree of $G$. True / False

**Problem 2 (20 points).** Given a directed graph $G = (V, E)$ with no cycle. Give a $O(|E| + |V|)$ time algorithm to check if there is a directed path that touches every vertex exactly once. Prove the correctness and the runtime of the algorithm.
Hints: You can use the fact that topological sort can be done in $O(|E| + |V|)$ time.

**Problem 3 (20 points).** Consider the interval scheduling problem in the class. We have $n$ jobs. The $j$-th job starts at $s(j)$ and finishes at $f(j)$. Jobs are compatible if they do not overlap. Our goal is to find a maximum subset of compatible jobs.

Instead of selecting the first compatible job to finish as in the class, we consider the greedy algorithm that selects the last compatible job to start. Prove that this yields an optimal solution or give an example to disprove this algorithm.

**Problem 4 (20 points).**
Given an array of positive numbers $a = [a_1, a_2, \cdots, a_n]$ . Give an $O(n \log n)$ time algorithm that find $i$ and $j$ (with $i \leq j$) that maximize the subarray product $\prod_{k=i}^{j} a_k$. Prove the correctness and the runtime of the algorithm.

For example, in the array $a = [3, 0.2, 5, 7, 0.4, 4, 0.01]$, the sub-array from $i = 3$ to $j = 6$ has the product $5 \times 7 \times 0.4 \times 4 = 56$ and no other sub-array contains elements that product to a value greater than 56. So, the answer for this input is $i = 3, j = 6$.

Hints: Divide and Conquer.