Santa Clara University
2021 Spring Final Exam

Course	COEN279/AMTH377	Name	
Department	Computer Engineering	Student ID	
Lecturer	Yuan Wang	Data/Time	2021/06/09, 7:00pm-9:00pm
Format	Open-book		
Note	No discussion		

1. True or False (13 questions, 2 points each, 26 points total)

1) Divide-and-Conquer technique can always lower the running time to be below $\Theta(n^2)$

2) Dynamic programming technique can improve the algorithm efficiency over brute force method by introducing more memory usage.

3) When using dynamic programming algorithm to solve a problem, each subproblem must have two or more subproblems.

4) Let X be a problem that belongs to the class NP. If X can be solved deterministically in polynomial time, then P = NP.

5) All NP-complete problems are NP-hard problems.

6) Since we are able to find NP-Complete problems, we know for sure that NP != P

7) For any problem X, if we can find a NP-complete problem C, and C is reducible to X, then X is also NP-Complete problem.

8) If an algorithm has a best case complexity $\Theta(m)$ and a worse case complexity $\Theta(n)$, then the complexity of the algorithm is $\Theta(m)$.

9) For CIRCUIT_SAT problem, if a circuit is not satisfiable, a verification algorithm is able to tell in polynomial time.

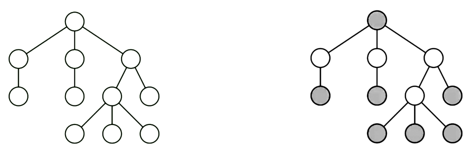
10) If any NP problem is not polynomial-time solvable, then no NP-Complete problem is polynomial-time solvable.

11) Let f(x) be a reduction function for $L1 \leq L2$. Since f(x) is an "if and only if" function, this also means that $L2 \leq L1$.

12) Optimal Binary Search Tree problem and Huffman Code problem are both finding optimal binary tree, since Huffman Code can be solved by Greedy Algorithm, Optimal BST problem can also be solved by Greedy Algorithm

13) Since Circuit_SAT ≤ Formula_SAT ≤ 3SAT ≤ CLIQUE ≤ VERTEX_COVER ≤ Hamilton_CYCLE, if we can find a polynomial algorithm for CLIQUE, then all NP problem and Circuit_SAT, Formula_SAT, 3SAT will be solved by polynomial algorithm, but not VERTEX_COVER and Hamilton_CYCLE

2. [30 points] We know that deciding whether an undirected graph has an independent set of size k is an NP-Complete problem. However, if this undirected graph is a tree, then finding maximum size of such a set is polynomial-time solvable. For example, the following tree (left) has a maximum independent set shown on the right.



a. Prove that all leaf nodes of the tree belong to a solution
b. Give a greedy algorithm to find the maximum size of this set
(You don't need to provide pseudocode, explain the idea and give main steps of actions of the algorithm)

3. [30 points] Following is a recursive relationship of substructure of a problem when being solved using dynamic programming. Give a top-down or a bottom-up algorithm in pseudo code.

$$C(0) = C(1) = 5$$

$$C(n) = \sum_{i=1}^{n-1} 2*C(i)*C(i-1), \text{ for } n>1$$

4. [14 points] In the proof of VERTEX_COVER problem being NP-complete, we reduce CLIQUE problem to VERTEX_COVER problem using complement as reduction function like this:

Explain why then do we say that CLIQUE problem is not harder to solve than VERTEX_COVER. Give necessary explanation of any statement you make.