Fall 2014 **CMSC 351: Final** Hamid Mahini
Time: 2 Hours

WAIT FOR INSTRUCTIONS BEFORE BEGINNING

HONOR PLEDGE: "I pledge on my honor that I have not given or received any unauthorized assistance on this examination."

Signature and UID: ____

Print name: ____

- ***Write your answers with enough detail about your approach and concepts used, so that the grader will be able to understand it easily. You should PROVE the correctness of your algorithms either directly or by referring to a proof in the book.***
- ***The sum of the grades is 110, but your grades would be out of 100 (thus you get 10 bonus points by solving all the problems).***
- ***Select the best choice for the first 5 problems and mark it by X in the table below***

Problem	1	2	3	4	5	6	7	8
a						X		
b		X			X			
c				X				
d			X				X	
e	X							X

DO NOT WRITE BELOW THIS LINE

Problem 1-8:	
Problem 9:	
Problem 10:	
Problem 11:	
Problem 12:	
TOTAL:	

Multiple-choice Problems (Answer ONLY in the TABLE in the FIRST PAGE and NOT HERE):

Problem 1. (**5 points**) Let $T(n) = 3T\left(\left\lfloor\frac{n}{2}\right\rfloor\right) + 5n$ and $T(0) = 7$. Then $T(n)$ is?

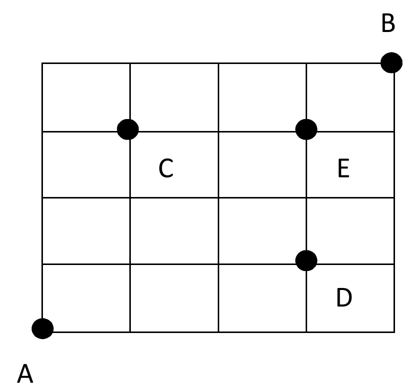
a) Θ(n)
b) $\Theta(n^{log_2^3}\log n)$
c) $\Theta(n\log n)$
d) $\Theta(n^{1.5})$
e) $\Theta(n^{log_2^3})$

Problem 2. (**5 points**) Let $f(n) = 3n^2 + logn$, $g(n) = nlogn + nlog^2n$, and $h(n) = 4n^2 - 5n$. Select the correct answer.

a) $f(n) = O(g(n)), f(n) = O(h(n))$
b) $f(n) = \Omega(g(n)), f(n) = \Theta(h(n))$
c) $f(n) = o(g(n)), f(n) = \Theta(h(n))$
d) $f(n) = \Theta(g(n)), f(n) = \Omega(h(n))$
e) $None$

Problem 3. (**5 points**) How many ways can you walk from A to B such you do not pass through C, D, and E. In each move you are allowed to go one step up or one step to the right. (Hint: use dynamic programming technique)

a) 8
b) 10
c) 12
d) 14
e) 18



Problem 4. (**5 points**) Problem A is polynomially reducible to problem B. Which of the following statements is correct?

I. If problem A is solvable in a polynomial time then problem B is solvable in polynomial time.

II. If problem A is NP-complete then problem B is NP-complete.

III. If problem A is NP-hard then problem B is NP-hard

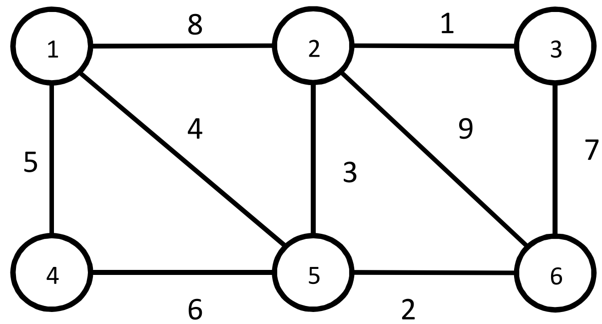
a) I
b) II
c) III
d) I and II
e) I, II, and III

Problem 5. (**5 points**) For dense graphs with $\Omega(n^2)$ weighted edges what is the best running time for Prim's algorithm?

a) $O(nlogn)$ b) $O(n^2)$ c) $O(n^2logn)$ d) $O(n^3)$ e) $O(nlog^2n)$

Problem 6. (5 points) Consider the weighted graph below. Run Dijkstra's algorithm starting from vertex 1. Write the vertices in the order which they are marked.

a) 1, 5, 4, 6, 2, 3
b) 1, 5, 4, 2, 6, 3
c) 1, 5, 6, 2, 4, 3
d) 1, 5, 4, 2, 3, 6
e) 1, 5, 6, 2, 3, 4



Problem 7. (5 points) Consider the weighted graph of Problem 6. Run Prim's algorithm starting from vertex 1. Write the vertices in the order which they are marked.

a) 1, 5, 4, 6, 2, 3

b) 1, 5, 2, 3, 6, 4

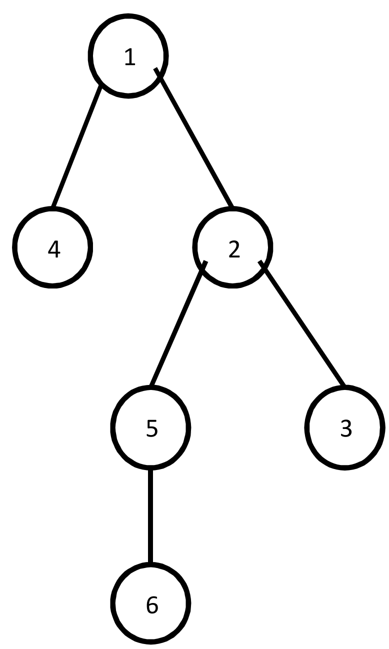
c) 1, 5, 6, 3, 2, 4

d) 1, 5, 6, 2, 3, 4

e) 1, 5, 4, 2, 6, 3

Problem 8. (5 points) Consider tree T below. Assume tree T is the DFS tree of undirected graph G. Which of the following cannot be an edge of graph G?

a) (1, 5)
b) (2, 6)
c) (1, 6)
d) (1, 3)
e) (4, 3)



Regular Problems (you should always PROVE THE CORRECTNESS of your solutions):

Problem 9. Given a sequence of (not necessary positive) integers $x_1, x_2, \ldots, x_n$ find a subsequence $x_i, x_{i+1}, \ldots, x_j$ (of consecutive elements) such that the sum of the numbers in it is **even** *and* **maximum** over all subsequences of consecutive elements with an even sum of elements. For example if the sequence is $5, -1, 4, -10, 3, 3, 3$, the maximum even consecutive subsequence is $5, -1, 4$. The running time of your algorithm should be in $O(n)$.

We say a subsequence of consecutive elements is "even" if the sum of elements in it is even, otherwise we say it is "odd". Define E_i as the maximum sum of even subsequences ending at i. Similarly, define O_i as the maximum sum of odd subsequences ending at i. Clearly the answer to the problem is $\max_i E_i$. We only need to show how we can compute E_is and O_is in $O(n)$. One can easily show that the following recursive formula holds, which would directly give us the desired algorithm: $E_0$and $O_0$are $-\infty$. For $i \geq 1$:

- If x_i is even => $E_i = \max\{x_i, E_{i-1} + x_i\}$ and $O_i = O_{i-1} + x_i$
- If x_i is odd => $E_i = O_{i-1} + x_i$ and $O_i = \max\{x_i, E_{i-1} + x_i\}$

Problem 10. Let T be a minimum spanning tree of a graph G, and let L be the sorted list of the edge weights of T. Prove that for any other minimum spanning tree T of G, the list L is also the sorted list of edge weights on T.

Consider two minimum spanning trees T and T', and let L and L' be the sorted list of the edge weights of T and T' respectively. Let $L = (w_{e_1}, w_{e_2}, \dots, w_{e_m})$ and $L' = (w_{e'_1}, w_{e'_2}, \dots, w_{e'_m})$. If $L = L'$ then we are done. Assume $L \neq L'$ and let i be the smallest index such that $w_{e_i} \neq w_{e'_i}$. Without loss of generality assume that $w_{e_i} < w_{e'_i}$. Add edge e_i to the minimum spanning tree T' and $T' + e_i$ will contain a cycle C. Look at all edges in cycle C. If the weight of all of them is less than or equal to w_{e_i}, then all of them would be in both L and L'. This means cycle C is in spanning tree T as well which is not possible. Therefore, the weight of at least one of edges in cycle C is greater that w_{e_i}. Let e'_jbe an edge of cycle C such that $w_{e_i} < w_{e'_j}$. Now consider spanning tree $T'' = T' + e_i - e'_j$. The sum of the edge weights of T'' is less that the sum of the edge weights of T'. This is a contradiction with the optimality of T'.

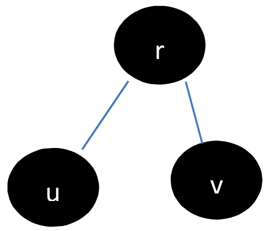
Problem 11. Given two sets S_1 and S_2 of natural numbers, and a natural number a. Find whether there exists an element x from S_1 and an element y from S_2 such that $2x + 5y = a$. The algorithm should run in time $O(n)$, where n is the total number of elements in both sets.

Use a Hash data structure to store all the members of S_2. For every member $x \in S_1$, we simply search for $\frac{a-2x}{5}$ in the data structure. In a hash data structure the running time of insertion and search are O(1) in average , thus the algorithm runs in O(n) time in average

Problem 12. Let $d[v]$ be the discovery time and $f[v]$ be the finishing time of vertex v in DFS. Prove that it is not possible to find two vertices v and u such that $d[u] < d[v] < f[u] < f[v]$.

$d[u] < d[v]$ means that we discover u before v. Now we have two cases either:

1. u is an ancestor of v
2. u is not an ancestor of v



If Case 1 happens then we know that every predecessor of u finishes before finishing u as DFS is a recursive function which contradicts with the fact that $f[u] < f[v]$.

If Case 2 happens then we know that u had to be finished by the discovery of v which contradicts with the fact that $d[v] < f[u]$.

Therefore the only valid sequences are : d[u] < d[v] < f[v] < f[u] or d[u] < f[u] < d[v] < f[v].

Problem 13. Explain Dijkstra's algorithm in details and prove its correctness.

Look at Section 24.3 of CLRS.