CS2223: Algorithms
D-Term, 2017
Midterm II
50 Minutes

Solution

<u>Question 1</u>

For each of the following sequences, indicate which one can be a valid search sequence in a binary search tree when searching for value 200.

If the sequence is valid, then write "Valid". If the sequence is not valid, then state the first incorrect (misplaced) value in the sequence.

<u>(a) [5 Points]</u> 70, 640, 320, 170, 210, 160, 200

This is not a valid sequence because value 160 is in the right sub-tree of value 170 which is wrong

<u>(b) [5 Points]</u> 420, 60, 300, 220, 50, 80, 200

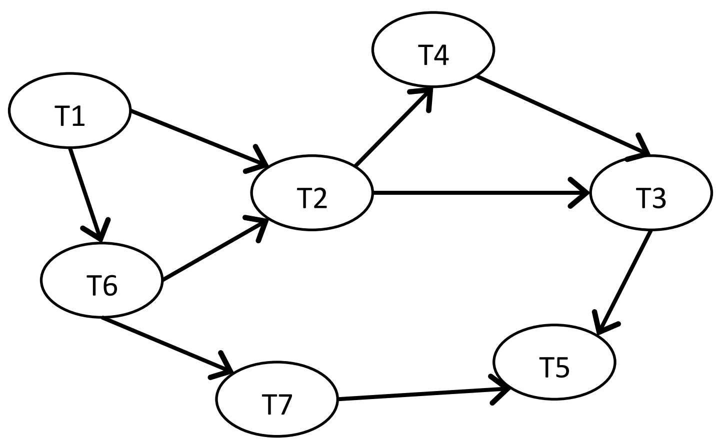
This is not a valid sequence because value 50 is in the right sub-tree of value 60 which is wrong

Question 2

Professor Alex is creating a set of topics for a newly developed course. The course will have seven topics T1, T2, …T7 with the following requirements:

1) T1 must be taught first.
2) T2 must be taught after T1 and T6, and T7 is after T6
3) T3 and T7 must be taught before T5
4) T2 must precede T3 and T4, and T4 must precede T3

(a) [5 Points] Create a graph that models the dependencies among the given topics.



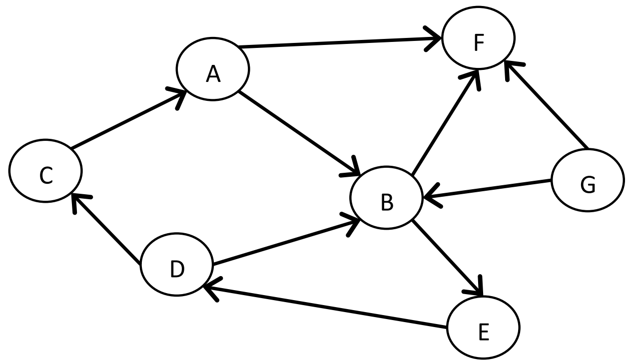
** If you added edge between T1 and all other topics, that will be also correct (although not necessary). It is necessary only to add from T1 to T6 to make sure T1 will be the first.

(b) [5 Points] Produce a teaching order for the given topics that Prof. Alex can follow and meets the above requirements.

T1,   T6,   T7,   T2,   T4,   T3,   T5

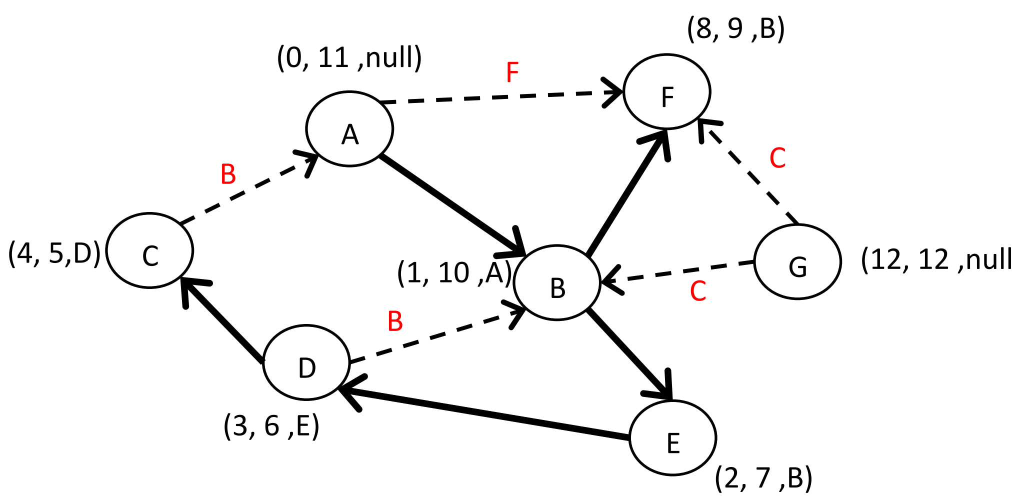
** There can be other orders that satisfy the requirements

<u>Question 3 [5 Points]</u>



Given the graph shown above. Apply DFS starting from node 'A'. Select the nodes of the out-going edges in alphabetical order. If you did not reach all nodes, then select any un-visited node and apply DFS again until all nodes are visited.

Re-draw the graph and mark each node with (startTime, endTime, Parent in the DFS tree). Also mark non-tree edges with either (B) for backward, (F) for forward, and (C) for cross edges.



Question 4

(a) [6 Points] Mark each statement as True or False

- DFS over an undirected graph cannot create any backward or cross edges

  False. Backward edges may exist, but forward and cross edges cannot

- Both BFS and DFS can be used to find the shortest path in un-weighted graphs in O(E+V)

  False. DFS cannot be used for shortest path.

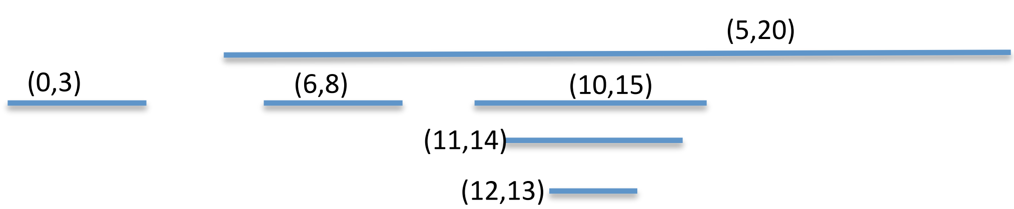
- Given an undirected weighted graph of 15 nodes, where the minimum edge weight is 3, then the total cost of any minimum spanning tree (sum of the tree edges) is greater than 32.

  True. MST will have 14 edges, since the minimum weight of each is 3, then the total sum will be greater than 32

(b) Assume someone applied DFS on a certain directed graph and marked each node with its (startTime, endTime) pairs. You are given a subset of these node labels (startTime, endTime) (not all node labels are given to you).

For each of the following subsets of node labels, indicate whether it is "Valid" (can be generated from DFS), or "Not Valid" (cannot be generated from DFS). In the latter case indicate why.

Case 1 [4 Points]: (5, 20), (6, 8), (0, 3), (10, 15), (11, 14), (12, 13)



"Not Valid"

Intervals do not partially overlap, which is good. However, if a node starts at time 6, it cannot end at time 8 because either it will go to another node, and in this case it will end at least at time 9. Or it will not go to other nodes, and in this case it will end at time 7.

Case 2 [4 Points]: (10, 15), (11, 25), (13, 18), (20, 24), (30, 35)

"Not Valid"

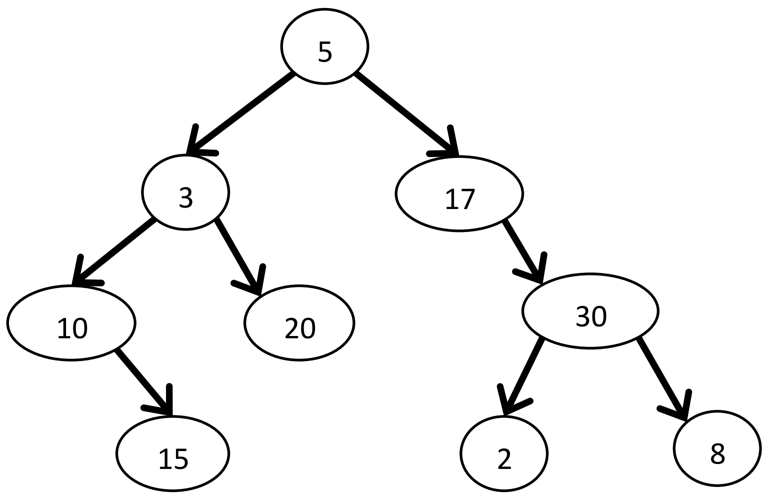
Because the intervals partially overlap such as (10, 15) and (11, 25) which cannot happen in DFS.

Question 5

(a) [5 Points] Given the following two traversals of a binary tree, re-construct (draw) the tree.

In-Order Traversal: 10, 15, 3, 20, 5, 17, 2, 30, 8

Post-Order Traversal: 15, 10, 20, 3, 2, 8, 30, 17, 5



Question 5

(b) [5 Points] Given a graph G= (V, E) and its minimum spanning tree T created by the Kruskal's algorithm. Assume the cost of one edge not in T is reduced. Write an efficient pseudocode to update T (the edge with the new cost may now be part of T).

1) Assume the modified edge become of cost c'
2) Remove from the tree T all edges with cost greater than c'
3) If none of the edges is removed, then terminate
4) Otherwise,
    - Apply Kruskal's algorithm over the edges with weights greater than or equal c'
    - If the modified edge is skipped, then you can immediately stop because the MST will be the same.
    - If the modified edge is selected, then keep applying the algorithm until the number of edges in T reaches |V| -1 again.