Final Exam

(Total Scores: 100)

Problem 1. Use the recursion tree technique to derive the upper bounds of the recurrences:
$T(n) = T(n-1) + 1$ (5 scores)

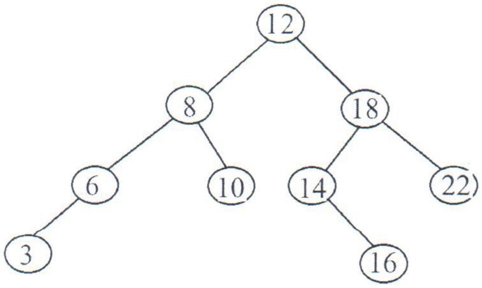
$T(n) = T\left(\frac{n}{2}\right) + 1$ (5 scores)

$T(n) = T(n-1) + T\left(\frac{n}{2}\right) + 1$ (5 scores)

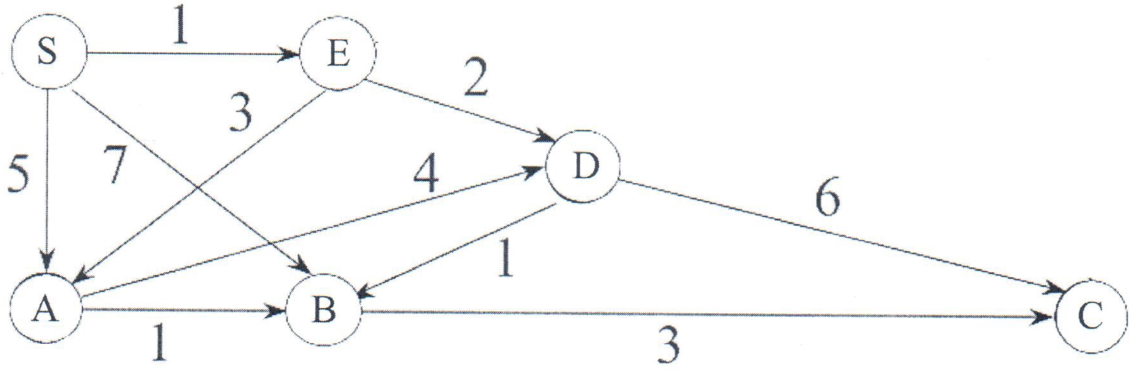
Problem 2. (a) Insert node with value 1 to the following balanced binary search tree (AVL) tree. Please show the position to insert the node. If the tree is not balanced after insertion, please rotate it to a balanced one. Please describe the rotation(s) such as right-rotation (node with value x) or left-rotation (node with value x). (5 scores)

(b) Delete node with value 22 from the current tree (after inserting 1). If the tree is not balanced after deletion, please rotate it to a balanced one. Please describe the rotation(s) such as right-rotation (node with value x) or left-rotation (node with value x). (5 scores)

(c) Delete node with value 12 from the current tree (after inserting 1 and deleting 22). If the tree is not balanced after deletion, please rotate it to a balanced one. Please describe the rotation(s) such as right-rotation (node with value x) or left-rotation (node with value x). (5 scores)



Problem 3. Run Dijkstra's algorithm and Bellman-Ford algorithm on the given graph shown below, respectively, starting at vertex S.



Dijkstra's algorithm: The second column shows the set S. The third to seventh columns show the shortest distances from s to other vertices. (10 scores)

	set S	A	B	C	D	E
initialization	S = {}	infinite	infinite	infinite	infinite	infinite
1st iteration						
2nd iteration						
3rd iteration						
4th iteration						
5th iteration						
6th iteration						

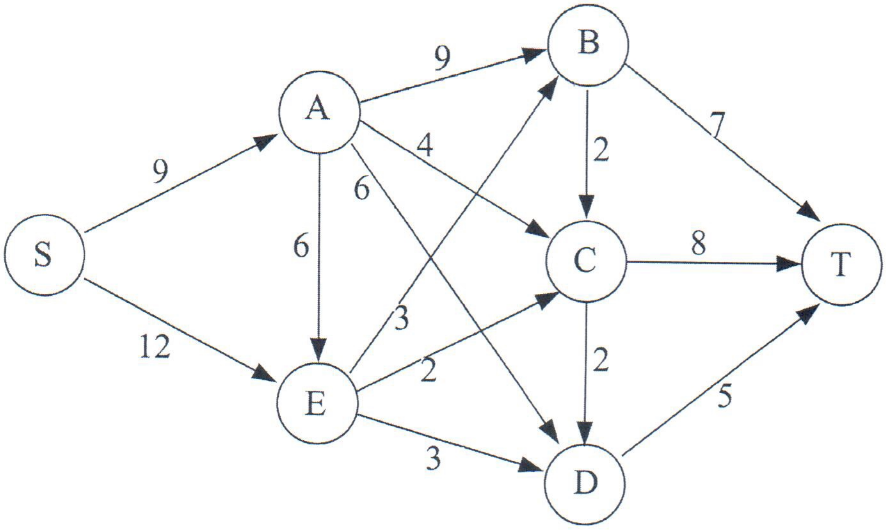
Bellman-Ford algorithm: The second to sixth columns show the shortest distances from s to other vertices. (10 scores)

	A	B	C	D	E
initialization	infinite	infinite	infinite	infinite	infinite
1st iteration					
2nd iteration					
3rd iteration					
4th iteration					
5th iteration					

Problem 4. Suppose you have an unlimited supply of three types of coins: d1 = 1 pennies, d2 = 2 pennies, and d3 = 3 pennies, please fill out the given table using a dynamic programming algorithm to find the fewest number of coins to make the change of 8. (10 scores) For the last entry of the table, please explain how the algorithm computes the value? (5 scores) Please give the exact coins making the fewest number of the change by checking the table, and explain. (5 scores)

0	1	2	3	4	5	6	7	8

Problem 5. Run the Ford-Fulkerson Algorithm to compute the max flow of the flow network shown below. Please draw the corresponding residual network at each step when running the Ford-Fulkerson Algorithm. (10 scores) Please compute the min-cut of the flow network. Please show the steps. (10 scores)



Problem 6. The following graph shows a sequence of courses with their dependencies on other courses. A directed edge from vertex u to vertex v indicates course u must be taken before course v can start. Please apply DFS (hint: topological sort) to find an ordering of these courses that conforms to the given dependencies. Please show each step. (10 scores)