Final: Spring 2019

Problem 1. Use the recursion tree technique to derive the upper bound of the recurrence: $T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{6}\right) + n$. (10 scores)

Problem 2. A knapsack with weight constraint W=10; four items with different values and weights listed in the first table. Suppose the maximal value has been computed as 90 based on dynamic programming (check the second table V[4,10]=90). Please explain the algorithm step by step to report the items selected to achieve the maximal value of knapsack based on the two tables. (10 scores)

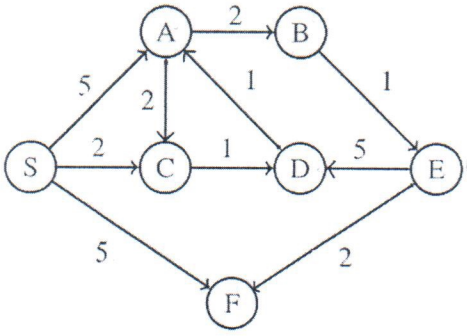
Let $W = 10$ and

i	1	2	3	4
v_i	10	40	30	50
w_i	5	4	6	3

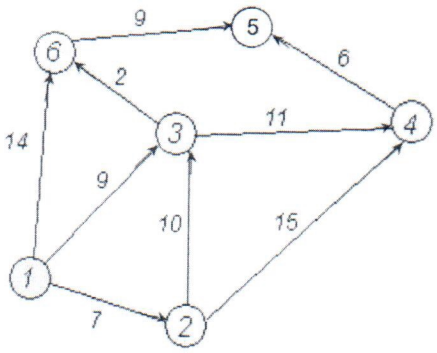
$V[i, w]$	0	1	2	3	4	5	6	7	8	9	10
$i = 0$	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	10	10	10	10	10	10
2	0	0	0	0	40	40	40	40	40	50	50
3	0	0	0	0	40	40	40	40	40	50	70
4	0	0	0	50	50	50	50	90	90	90	90

Problem 3. Knapsack problem belongs to NP-Complete Problem. However, it seems that the problem can be efficiently solved using dynamic programming as the previous one shown. Please help clarify the confusion. (10 scores)

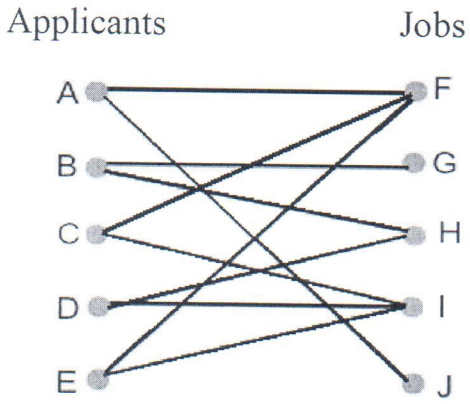
Problem 4. BFS/DFS: Give the visited node order for each type of graph search, starting with s. (10 scores) Please apply DFS to identify cycles of the given graph (10 scores) and remove cycles by deleting the minimum number of edges (5 scores). Please topologically sort the directed acyclic graph (after removing cycles) and show it in sorted order (10 scores).



Problem 5. Find the shortest path from node 1 to all other nodes using Dijkstra's and Bellman-Ford algorithms on the given graph shown below, respectively. Please show each step. (15 scores)



Problem 6. The bipartite graph shows the possible allocation of people to jobs. Use Ford-Fulkerson algorithm to find a maximal matching. Please show each step. (20 scores)



Bonus question: There are two prisons with N prisoners in total in city S. Every pair of prisoners has a possibility of fighting with each other. Suppose the N prisoners are labeled from 1 to N with fighting possibility of every pair given. Please design an algorithm to divide the N prisoners to the two prisons to minimize the total fighting possibilities. (30 scores)