Algorithm 2016 fall

Quiz

1. (10%) Illustrate the **BREATH-FIRST SEARCH** algorithm and analysis its time complexity.

2. (10%) Find the strongly connected components on a directed graph in Figure 1.

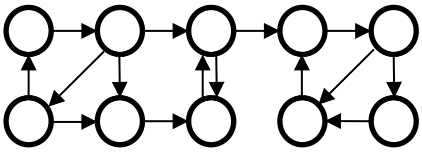

Figure 1

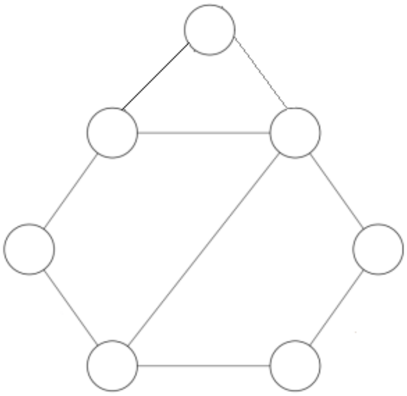

Figure 2

3. (20%) (1) (10%) The matrix-chain multiplication problem can be stated as follows: Given a chain $<A_1, A_2, ..., A_n>$ of n matrices, where for i=1,2,...,n, matrix A_i has dimension $p_{i-1} \times p_i$, fully parenthesize the product $A_1A_2......A_n$ in a way that minimizes the number of scalar multiplications. Give a dynamic-programming algorithm to solve the problem and analyze its time complexity. (write pseudo code)

   (2) (10%) Suppose that you have 6 matrices: *A1* has dimension 30 x 35, *A2* has dimension 35 x 15, *A3* has dimension 15 x 5, *A4* has dimension 5 x 10, *A5* has dimension 10 x 20, *A6* has dimension 20 x 25. Please use your algorithm to calculate the minimum number of scalar multiplications.(write table and answer)

4. In the algorithm SELECT, the input elements are divided into groups of 5. Will the algorithm work in linear time if they are divided into groups of 7? Argue that SELECT does not run in linear time if groups of 3 are used.

5. (10%) Using the depth-first-search algorithm DFS on an undirected graph in Figure 2. Vertices are timestamped by discovery time/finishing time. (draw your answer)

6. Consider the knapsack problem consists of 3 items, and the capacity of the knapsack is equal to 8. The profits and weights of the three items are (p1, p2, p3) = (8, 6, 3) and (w1, w2, w3) = (6, 5, 3), respectively.
   Assume that you are allowed to put in a fraction of an item. Use the greedy method to solve for the maximum profit and show the items to be included in the knapsack.

7. Consider the following sentence:
   In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub-problems.
   Is it correct? Please give a short answer briefly (you can give an example to explain your answer)

8. True or false
   (I) 0-1 knapsack problem can be solved by greedy strategy
   (II) In undirected graph, we apply DFS algorithm and we can get tree edges and cross edges
   (III) In top-down approach, memorization is a way to reduce some time cost in exchange for space cost
   (IV) Unweighted longest simple path problem does not satisfy the optimal sub-structure
   (V) If we apply topology sort for a directed graph, then using DFS on this graph will produce no back edges

9. Consider the following table.

item i	1	2	3	4	5	6	7
$value_i$	8	6	15	3	2	5	9
$weight_i$	2	3	5	4	3	2	6

   Imaging that you have a travel and the maximum weight of your knapsack is 16. Now, you cannot take any fraction of an item. It means that you cannot take any item partially. What is the optimal value of items that you can choose into your knapsack?