National University of Computer & Emerging Sciences, Karachi

Fall-2020 Department of Computer Science

Mid Term-2

24th November 2020, 10:30 AM – 12:00 PM

Course Code: CS302	**Course Name:** Design and Analysis of Algorithm
Instructor Name / Names: Dr. Muhammad Atif Tahir, Dr. Fahad Sherwani, Zeshan Khan, Waqas Sheikh, Sohail Afzal	
Student Roll No:	**Section:**

Instructions:

- Return the question paper.
- Read each question completely before answering it. There are **6 questions** on **3 pages.**
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.

Time: 90 minutes. **Max Marks: 17.5**

Question # 1 **[3 marks]**

Consider the following instance of the 0/1 knapsack problem

items	1	2	3	4
values	3	4	5	6
weights	2	3	4	5

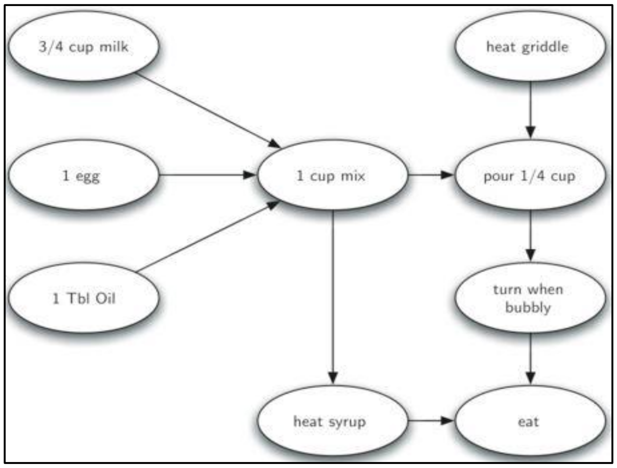
The maximum allowable total weight in the knapsack is W = 5.

Find an optimal solution for the above problem with the weights and values given above using Dynamic Programming. Be sure to state both the maximum value as well as the item(s) that will make maximum value in the given capacity W. Show all steps.

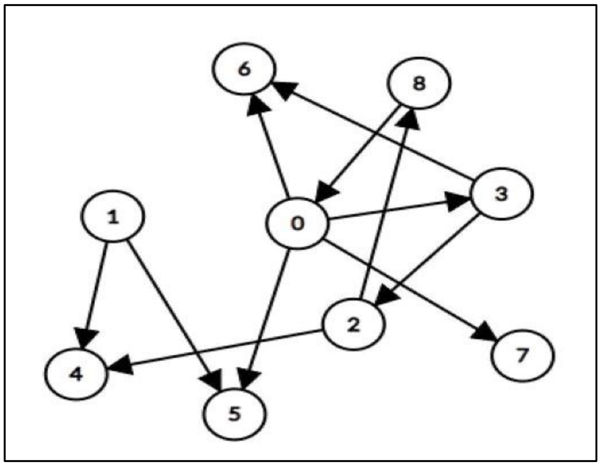
Question # 2 **[1+0.5 = 1.5 marks]**

Given two directed graphs in part (a) and part (b), you need to write one topological ordering of both graphs if possible.

Part (a)



Part (b)



Question # 3 **[1 + 1 = 2 marks]**

Compute the time complexity for both below mentioned algorithms. Show all steps.

```
void Algorithm1(A, int n){
        for(int i = 0; i < $\sqrt{n}$; i++){
                for (int j = 0; j < $\sqrt{n}$; j++){
                        int temp = A[j][i]/A[i][i]
                        for (int k = i + 1; k ≤ n + i; k++){
                                A[j][k − i]−= temp ∗ A[i][k − i]
                        }
                }
        }
}
```

```
void Algorithm2(A, int n){
        for(int i = 0; i < n/2; i++){
                for (int j = 1; j ≤ n/2; j++){
                        for (int k = i + 1; k ≤ i + 5; k++){
                             for (int l = 1; l ≤ n; l++){
                                        sum+= A[i + j][l]
                        }
                }
        }
}
```

<u>Question # 4</u> **[3 marks]**

Let A and B be two sequences of *“n”* integers each, in the range $[1, n^2]$, from square series. Given an integer *x*, design an O(n)-time algorithm for determining if there is an integer “*a”* in A and an integer “*b”* in B such that *x* = $a + b$. (You may write algorithm in plain text)

<u>Question # 5</u> **[4 marks]**

Given a graph *G* and a minimum spanning tree *T*, suppose that we decrease the weight of one of the edges not in *T*. Give an algorithm for finding the minimum spanning tree in the modified graph. (You may write algorithm in plain text)

<u>Question # 6</u> **[0.5 + 0.5 + 3 = 4 marks]**

Answer the following.

a) How do we decide to split the matrix-chain and parenthesize for the optimization of multiplications operations?

b) Which of the following is the recurrence relation for the matrix-chain multiplication problem where mat[i-1] * mat[i] gives the dimension of the ith matrix?

1) M[i, j] = 1 if i=j
M[i, j] = $\min\limits_{i \le k < j}$ { M[i, k] + M[k+1, j]}

2) M [i, j] = 0 if i=j
M [i, j] = $\min\limits_{i \le k < j}$ {M[i, k] + M[k+1, j]}

3) M [i, j] = 0 if i=j
M [i, j] = $\max\limits_{i \le k < j}$ {M[i, k] + M[k+1, j]} + mat[i-1]*mat[k]*mat[j]

4) M [i, j] = 0 if i=j
M [i, j] = $\min\limits_{i \le k < j} \{\mathrm{M}[\mathrm{i},\mathrm{k}] + \mathrm{M}[\mathrm{k}+1,\mathrm{j}]\} + \mathrm{mat}[\mathrm{i}-1] * \mathrm{mat}[\mathrm{k}] * \mathrm{mat}[\mathrm{j}]\}$

c) Design a **recursive algorithm** to compute the minimum number of scalar multiplications for the chain matrix product $A_{i...j}$. (You may write algorithm in plain text)

<u>BEST OF LUCK</u>