National University of Computer & Emerging Sciences, Karachi
Fall-2022 Department of Computer Science
Mid Term-2
03 November 2022, 11:30 AM – 01:00 PM

Course Code: CS2009	**Course Name:** Design and Analysis of Algorithm
Instructor Name / Names: Dr. Muhammad Atif Tahir, Dr. Farrukh Saleem, Dr. Waheed Ahmed, Miss Anaum Hamid, Miss Aqsa Zahid, Sohail Afzal	
Student Roll No:	**Section:**

Instructions:

- Return the question paper.
- Read each question completely before answering it. There are **6 questions** on **4 pages.**
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.

Time: 90 min **Max Marks: 17.5**

Question # 1 **(15 min)** **[3 marks] [CLO 4]**

Find the Shortest Common Super sequence (SCS) for the strings: X=NOOR, Y=ABDULLAH. First dry run to find the solution of Longest Common Subsequence (LCS) and then further dry run the below algorithm to find the SCS.

```
//Let m and n contain the length of strings X and Y,
and the matrix dp[m + 1][n + 1] contains the LCS
solution.

string printShortestSuperSeq(string X, string Y) {
   string str;
   int i = m, j = n;
   while (i > 0 && j > 0) {
      if (X[i - 1] == Y[j - 1]) {
         str.push_back(X[i - 1]); i--, j--; }
       else if (dp[i - 1][j] > dp[i][j - 1]) {
         str.push_back(Y[j - 1]); j--; }
      else{
         str.push_back(X[i - 1]); i--; } }

   while (i > 0) {
      str.push_back(X[i - 1]); i--; }
   while (j > 0) {
      str.push_back(Y[j - 1]); j--; }

   reverse(str.begin(), str.end());
   return str;}
```

Question # 2 **(5 min)** **[2 marks] [CLO 1]**

Suppose we have two different searching algorithms having complexity of $2n^2$ and $100n$. We are running $2n^2$ complexity algorithm on Computer A (Intel i7 10-core) which executes 200 billion instruction per second and we are running $100n$ complexity algorithm on Computer B (Intel i860) which execute only 50 million instruction per second. Suppose Input length is 1 billion; identify the time taken by Computer A and Computer B.

Question # 3 **(15 min)** **[2.5 marks] [CLO 3]**

a) Suppose we are given two unsorted integers sequences *A* and *B*, which may contain duplicate entries. Design an ***O(n)*** time algorithm for computing an integer sequence representing the set AUB (with no duplicates) where U stands for union.

> **Example:** Suppose A is [22, 350, 1, 350, 22, 350, 1] and B is [31, 13, 350, 35, 111, 22]. The resulting list we would like to have is [1, 13, 22, 31, 111, 350]

b) Discuss space complexity of proposed algorithm briefly

Question # 4 **(10 min)** **[1+1 = 2 marks] [CLO 4]**

1) Figure (a) shows graph of game of Pick up Sticks. An edge indicates that one sticks overlaps another. Write valid order for picking up the sticks
2) Figure (b) shows recipe steps to make pasta. Write valid order for making this dish

Figure (a)

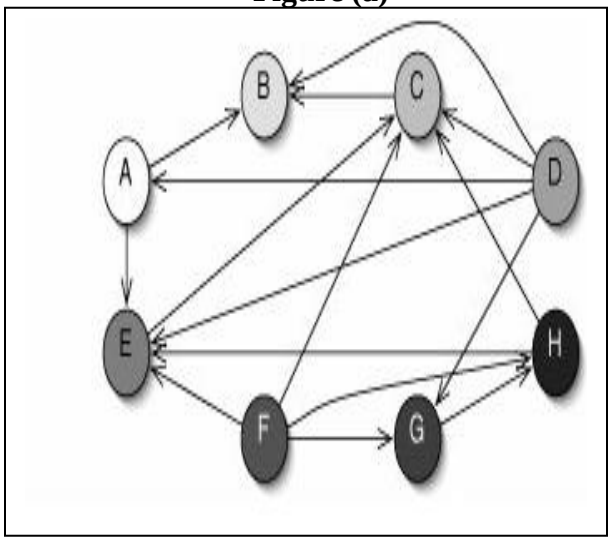
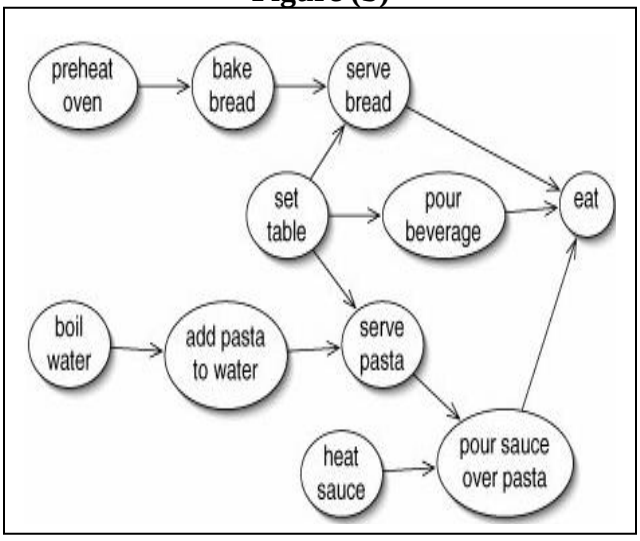


Figure (b)



Question # 5 **(15 min)** **[2.5 marks] [CLO 3]**

You need to dry run the following algorithm to show all the necessary steps to sort an array A where array A= (455, 61, 63, 45, 67, 135, 74, 49, 15, 5) and d= 3. Show all steps clearly.

```
1  Sort(A, d)
2          for j = 1 to d do
3              int count[10] = {0};
4              for i = 0 to n do
5                  count[key of(A[i]) in pass j]++
6              for k = 1 to 10 do
7                  count[k] = count[k] + count[k-1]
8              for i = n-1 downto 0 do
9                  result[ count[key of(A[i])] ] = A[j]
10                 count[key of(A[i])]--
11             for i=0 to n do
12                 A[i] = result[i]
13         end for(j)
14  end func
```

Question # 6 **(30 min)** **[1+1.5+3 = 5.5 marks] [CLO 4]**

Suppose you're running a lightweight consulting business — just you, two associates, and some rented equipment. Your clients are distributed between Karachi and Lahore, and this leads to the following question.

Each month, you can either run your business from an office in Karachi (KH) or from an office in Lahore (LR). In month ***i***, you'll incur an operating cost of $\mathbf{K_i}$ if you run the business out of KH; you'll incur an operating cost of $\mathbf{L_i}$ if you run the business out of LR. The costs depend on the distribution of client demands for that month. However, if you run the business out of one city in month ***i***, and then out of the other city in month ***i + 1***, then you incur a fixed moving cost of ***M*** to switch base offices.

Given a sequence of ***n*** months, a plan is a sequence of ***n*** locations—each one equal to either KH or LR—such that the $\boldsymbol{i^{th}}$ location indicates the city in which you will be based in the $\boldsymbol{i^{th}}$ month. The cost of a plan is the sum of the operating costs for each of the **n** months, plus a moving cost of ***M*** for each time you switch cities. The plan can begin in either city.

The problem. Given a value for the moving cost M, and sequences of operating costs $K_1, \ldots, K_n$ and $L_1, \ldots, L_n$, find a plan of minimum cost. (Such a plan will be called optimal)

Example. Suppose n = 4, M = 10, and the operating costs are given by the following table.

	Month 1	Month 2	Month 3	Month 4
K_i	1	3	20	30
L_i	50	20	2	4

Then the plan of minimum cost would be the sequence of locations [KH, KH, LR, LR], with a total cost of 1 + 3 + 2 + 4 + 10 = 20, where the final term of 10 is the moving cost of changing locations once.

a) Show that the following algorithm does not correctly solve this problem, by giving an instance (example) on which it does not return the correct answer. Assume that n = 4 and M = 10 as it was in the example above. Give the optimal solution and its cost, as well as what the algorithm finds.

```
for i = 1 to n
        if K_i < L_i then
                output "KH in Month i"
        else
                output "LR in Month i"
end
```

b) Give an example of an instance (again with n = 4 and M = 10) in which the optimal plan must move (i.e., change locations) at least three times. Provide a brief explanation, saying why your example has this property.

c) Give a pseudo code for an efficient dynamic programming algorithm with complexity O(n) that takes values for n, M, and sequences of operating costs K1, . . . , Kn and L1, . . . , Ln, and returns the cost of an optimal plan.
Hint: What is the table of intermediate results with optimal substructure property?

The End