United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Midterm Exam Total Marks: **30** Summer-2023

Course Code: CSE2217 Course Title: Data Structure and Algorithms II

Time: *1 hour 45 minutes*

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

There are **Four** questions. **Answer all of them**. Show all the calculations/steps, where applicable. Figures in the right-hand margin indicate full marks.

1 **(a)** Derive the **best-case**, and the **worst-case** running time equations for the following function *connectDots* and represent using Asymptotic Notation. **[4]**

```
1 bool connectDots(int arr[], int n){
2     int numberOfDots = 0;
3     for(int i=0; i<n; i++){
4         if(arr[i]&1){
5             for(int j=i+1; j<=n-1; j++){
6                 numberOfDots++;
7             }
8             for(int j=0;j<=i; j++){
9                 numberOfDots *= 2;
10             }
11         }
12         int j = 1;
13         while(j<=n){
14             numberOfDots+=j;
15             j = j * 2;
16         }
17     }
18     return (numberOfDots&1);
19 }
```

(b) Derive the **exact-cost equation** for the running time of the following function and find the time complexity in big-oh notation. **[4]**

```
1 for (int i = 1; i <= n; i = i * 2){
2     for (int j = 1; j <= i; j++){
3         for (int k = n; k >= i; k--){
4             printf("%d ", k);
5         }
6         printf("\n");
7     }
8     printf("\n");
9 }
```

2 **(a)** Solve the following **recurrence equation**, where T(1) = O(1). **[2]**

$$T(n) = 4T(n/2) + O(n)$$

(b) You are given an array of integers A = {1, -3, 2, 1, -1, 4, -2, 3, -1, 2, -3, 4}, find the **maximum sum subarray** using divide-and-conquer approach. You must show the recursion tree and clearly mention left, right and crossing sum for each tree node. **[3]**

(c) Suppose we have two sorted sub-arrays: L: 1, 5, 7, 8, 10, 12 and R: 4, 6, 7, 9, 13, 14. Perform the **procedure Merge** on L and R to find the final sorted array A. Show each step of your answer and the number of comparisons required in each step. **[2]**

3 **(a)** What is **optimal substructure** property? Write down the optimal substructure property of the **coin change** problem. **[1+1]**

(b) Demonstrate why the recursive approach to calculate a Fibonacci number is inefficient, by calculating the Fibonacci number F_5. How does the dynamic programming approach for the same solve this inefficiency? (Consider $F_0 = 0, F_1 = 1$) **[1+1]**

(c) A smuggler enters a warehouse to find the items listed in the following table. He has a bag to carry the smuggled goods, but it can carry only 8 kg weight at best. The smuggler wants to leave with the items that will result in a maximum profit for him. Note that he cannot take an item partially; he either will take the item, or will not. **[4]**

Using dynamic programming, **calculate the maximum profit** the smuggler can earn.

Item no.	**1**	**2**	**3**	**4**
Weight	3	5	4	6
Profit	10	30	25	50

4 **(a)** Following items are available in a grocery shop: **[3]**

- 12 kilogram rice grain which costs 840 taka
- 10 kilogram salt which costs 870 taka
- 8 kilogram saffron powder which costs 2000 taka and
- 5 kilogram sugar which costs 500 taka

A group of thieves (Thief 1, Thief 2, … Thief M) have come to steal from that shop, **each with a knapsack of capacity 9 kg**. The thieves are entering in serial, *Thief 2* enters after *Thief 1* is done with stealing, *Thief 3* enters after *Thief 2* is done with stealing and so on. *Since each thief wants to maximize his/her profit*, **how many thieves** will be needed in the group to empty the grocery shop and **what are the items** that each of those thieves carry? Show details of the calculation.

(b) A document to be transmitted over the internet contains the following characters with their associated frequencies as shown in the following table:

Character	a	e	l	n	o	s	t
Frequency	74	105	44	55	73	57	49

Use Huffman technique to answer the following questions:

i. Build the Huffman code tree for the message and find the codeword for each character. **Encode** "*stolen*" using the codewords. **[3]**

	ii.	What is the **percentage saving** if the data is sent with fixed-length code values without compression?	**[1]**