United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Midterm Exam Total Marks: **30** Spring-2024

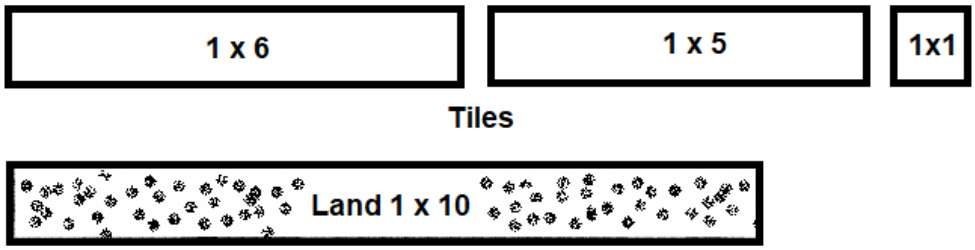
Course Code: CSE2217 Course Title: Data Structure and Algorithms II

Time: *1 hour 30 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)** Imagine you have a land with an area of 1x10 m^2, and you want to cover it using some tiles. There are three types of tiles available which varies in size: 1x1 m^2, 1x5 m^2 and 1x6 m^2 (See figure). Each type has infinite amount of supplies, so you never run out of tiles. Now, there are many ways that you can cover that land using these tiles, but you want to use as few tiles as possible (Tiles are expensive!) **[4]**



Using ***Dynamic Programming*** method, find the minimum number of tiles that can cover your land. Which tiles should we use? Describe your solution with detailed calculation.

(b) Suppose your wallet has capacity to hold only 8 grams of gold coins, and your best friend just offered you 4 gold coins from his own collection. The weights and the values of the coins are as follows: [5g, 4g, 6g, 3g] and [110$, 100$, 120$, 90$]. Using ***Dynamic Programming,*** determine which coins you should take so that your total gain is maximized. Keep in mind that you cannot carry more than 8 grams of gold coins. **[4]**

2 **(a)** Consider the following array **[4]**

-1	10	-1	6	-2	-4	2	2	8	-4

Find the Maximum-sum subarray using divide-and-conquer. You must show the recursion tree and clearly mention left, right and crossing sum for each tree node.

(b) Design an algorithm using the divide and conquer approach that counts the number of elements in an array that end with the digit '1'. The algorithm should make use of the modulus operator to determine if an element ends with '1'. Include the pseudocode for your algorithm, clearly defining the base case and detailing all necessary steps and calculations. **[3]**

(Please Turn Over)

3	**(a)** Using recursion tree method or the Master Theorem find out a good asymptotic upper bound on the following recurrence: $T(n) = 3T(n/2) + O(n)$ **(b)** Consider the following function which takes two arrays (A and B) and their lengths (n and m) as inputs respectively. Now determine the following: i. The exact-cost equation for the running-time. ii. The best case and worst case running time using the Big - Oh notation iii. Examples of best case and worst case inputs	**[3]** **[1+2+ 2 = 5]**
4	**(a)** Given the arrival and the departure times (in minutes) of 8 trains for a railway platform, **find out the maximum number** of trains that can use that platform without any collision, using a **greedy algorithm**. There must exist at least 10 minutes of safety break between the departure of one train and arrival of the next one. [1000, 1030], [840, 1030], [850, 1040], [1700, 2000], [800, 835], [1300, 1800], [1500, 1650], [1200, 1380] **Explain** your strategy very briefly and show detailed calculations. *No need to write pseudocode*. **Explain** why your solution satisfies the *optimal substructure property*. **(b)** "Data encoded using Huffman coding is uniquely decodable"- is the statement true or false? **Justify** your answer. (Please Turn Over)	**[2+1]** **[1]**

```
void func1 (int A[], int B[], int n, int m){
    int sumA = 0;
    for(int i=0;i<n;i++){
        sumA = sumA + A[i];
    }

    int sumB = 0;
    for(int i=0;i<m;i++){
        sumB = sumB + B[i];
    }

    int count = 0;

    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            if(A[i]*A[i]>B[i] || sumA>sumB){
                count++;
            }else{
                break;
            }
        }
    }
}
```

(c) Suppose you have a cup with capacity 750 ml. The following table shows that there are 5 flavors of drinks. For each flavor, there is only *d* ml available, and each flavor of drink contains a total of *s* grams sugar. **At most how many grams of sugar** can you consume if you fill up the cup using **greedy approach**? You need to show all the steps of your calculation. **[3]**

Flavor	Volume (*d*)	Total sugar (*s*)
Apple	320	35
Orange	220	20
Pineapple	240	40
Cranberry	200	30
Strawberry	280	25

(End of paper)