ourdiu.com | Shared by Hasibur Rahman

Daffodil International University

Faculty of Science & Information Technology
Department of Computer Science & Engineering
Mid Examination, Spring 2025
Course Code: CSE213, Course Title: Algorithms
Level: 2 Term: 1 Batch: 66

Time: 01:30 Hrs **Marks: 25**

Answer ALL Questions

[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]

1. Analyze the complexity of the following code: — **Marks** — **CO1**

a)

```
for(int i = 1; i <= n; i++) {
    for(int j = 1; j <= n; j *= 2) {
        int x = 5;
        printf("i = %d, j = %d\n", i, j);
    }
}
for(int i = 1; i <= n; i++) {
    printf("i = %d\n", i);
}
```

b)

```
void exampleFunction(int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("*");
        }
    }
}
```

2.5 + 2.5

2. Consider a **sorted dataset** containing **1 million** unique integers, ranging from **1 to 1,000,000**. — **05** — **CO2**

a) If you perform a **binary search** to find a randomly chosen number, how many comparisons would it take in the worst case?
b) If you perform a **linear search** instead, how many comparisons would it take in the worst case?
c) Given the time complexity of both algorithms, how would the performance differ if the dataset size **doubled** to **2 million**?
d) Suppose the dataset is **unsorted.** Which search algorithm would be preferable and why?
e) If the target number is found early in the list, which search method would be more efficient, and under what conditions?

3. You are given an unsorted array of integers: **[4, 7, 5, 9, 1, 3, 8, 6]**. Your task is to sort this array in descending order using a divide-and-conquer sorting algorithm. Additionally, you must ensure that the algorithm does not use any extra memory (i.e., it sorts in place) and **display the intermediate steps** of the sorting process as the array is sorted to help demonstrate how the algorithm works. — **05** — **CO2**

4. Given the message: **khokakhokikhokamaniratakccare**
Construct a Huffman tree, assign variable-length codes, and encode the message. Compute the total bits required. — **05** — **CO3**

ourdiu.com | Shared by Hasibur Rahman

5.	Imagine you're packing for a weekend trip, and you have a limited amount of space in your backpack. You have **6 items** to choose from, and each item has a **weight** (how much space it takes up in your backpack) and a **value** (how important or useful it is for your trip). You want to pack the most valuable combination of items, but your backpack has a weight limit of **11 kg**. You can either take an item with you or leave it behind, but you can't take the same item more than once. Find the most valuable combination of items from the following list and mentioned the name of the items.	**05**	**CO3**

Product	Weight	Value
Book	2	3
Camera	3	4
Pair of shoes	4	5
Jacket	5	6
Portable charger	6	7
Blanket	7	8

ourdiu.com

B→5