Daffodil International University

Faculty of Science & Information Technology

Department of Software Engineering

Midterm Examination, Summer 2026

Course Code: SE214; Course Title: Algorithm Design and Analysis

Sections & Teachers: AS (A, B, C), MMSI (D, E), AAS (F, G), AF (H, I), SA (J)

Time: 1 Hour 30 Mins **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.	***a)***	While developing a solution for a computational problem, a software engineer must ensure that the chosen algorithm is both correct and efficient. **Explain** the concept of an algorithm and **analyze** the essential characteristics that define a good algorithm.	**[Marks-2]**	**CLO-1** *Level-4*
	b)	**Break down** the execution of the following code snippet and **identify** its time complexity.	**[Marks-3]**	
2.	***a)***	A weather station at Daffodil International University recorded the following temperature readings (in °C) during a winter day: **[15, −8, 22, −12, 4, 0, −5]** To make the data easier to analyze, the meteorological team wants to arrange the temperature readings from the lowest value to the highest value. The sorting process begins by considering the first reading as already arranged, and then each subsequent reading is inserted into its correct position among the previously arranged readings. **Explain** the step-by-step process used to arrange the temperature readings. Finally, provide the sorted sequence and briefly **explain** the best-case and worst-case time complexities of this sorting approach with proper justification.	**[Marks-4]**	**CLO-2** *Level-2*

```c
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    for (int i = 1; i < n; i++) {
        for (int j = 0; j < i; j += 4) {
            printf("Done\n");
        }
    }
    return 0;
}
```

```c
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    int count = 0;
    for (int i = 1; i < n; i *= 2) {
        count++;
    }
    for (int j = 0; j < n; j++) {
        count++;
    }
    printf("%d\n", count);
    return 0;
}
```

	b)	During the annual admission process at Daffodil International University, the registration system received the following student application IDs in a random order: **[38, 12, 45, 7, 29, 18, 50, 3, 41, 24]** Due to the large number of records, the system organizes the data by handling smaller portions of the list independently before combining the results into a final ordered sequence. The goal is to arrange the application IDs from the largest value to the smallest value. **Visualize** each step involved in organizing the application IDs and show how the final sequence is obtained.	**[Marks-4]**	
	c)	A university is organizing a technology festival and has allocated **700 square meters** of space for corporate exhibition stalls. Several companies have applied for stalls, each requiring a certain area and offering a sponsorship amount in return .	**[Marks-7]**	

Company	Area Required (sq. m)	Sponsorship (Tk.)
A	120	84,000
B	180	144,000
C	150	90,000
D	200	180,000
E	250	250,000
F	100	50,000

The organizing committee aims to maximize the total sponsorship without exceeding the available stall space. Two greedy strategies are considered:
(i) selecting companies with the minimum area requirement first and
(ii) selecting companies with the highest sponsorship-to-area ratio first

Compare both strategies and **distinguish** which strategy provides a better allocation. Briefly **explain** the greedy-choice principle and **discuss** why that strategy performs better in this scenario.

3.	*a)*	During an inter-university programming competition, participants were required to exchange short text messages with their teammates while minimizing network bandwidth usage. One team decided to use Huffman Coding, a greedy algorithm, to compress their messages before transmission. To test their implementation, they encoded the text message **"SUCCESSWINNER"** **Illustrate** the Huffman Tree for the given message and determine the Huffman code for each character.	**[Marks-5]**	**CLO-3** ***Level-4***