Daffodil International University

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Midterm Examination, Fall 2022
Course Code: CSE214 Course Title: Algorithm
Level: 2 Term: 1 Batch: 58, 59, Old Syllabus

Time: 1.5 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. a) Analyze the Big-Oh(O) complexity of the following two code snippets: — CO1

i)

```
for( i = 1 ; i < n ; i++)
  {
      for( j = 1 ; j <= i ; j++)
      {
          printf("Algorithm is fun");
      }
      printf("\n");
  }
```

[2]

ii)

```
for( i = 1 ; i <= n ; i++)
      for ( j = 1 ; j <= n ; j++)
          for ( k = 1 ; k <= j ; k++)
              printf("I am a genius!\n");


  for( i = 1 ; i <= n ; i++)
      printf("I am again a genius!\n");
```

[2]

b) Consider the following algorithm (on the next page) and the given array. Now run the algorithm on the array and demonstrate how the array will be updated at each iteration. Show every single step. [3]

A[5] = {4, 7 ,2 ,6, 8} and n=5

```
for( i=0; i<=n/2; i++)
 {
   for( j=i; j<=i; j++)
      {
        k=A[i];
        A[i]=A[n-i-1];
        A[n-i-1] = k;
      }
 }
```

2.	Identify which searching algorithm you will be using in each of the following scenarios? Explain your choice (in no more than 3 lines per scenario).		**[6]**	CO2
	i.	You have an array of 10 million floating point numbers sorted in descending order. You want to find a number from this array – let's call this operation a query. There is only a single query in this scenario.		
	ii.	Now, you have the same array of same size but this time they are in ascending order. Again, there will only be a single query.		
	iii.	For the third scenario, the numbers are unsorted. Again, there will be a single query only.		
	iv.	For the fourth scenario, the numbers are unsorted and you have to make 1 million different queries.		
3.	Both merge sort and quick sort algorithms follow divide and conquer approach and their run time complexity in average case is **O(n log n)**. Considering the following list, compare the performance of merge sort and quick sort algorithm for sorting the following list. [7 6 5 4 3 2 1]		**[5]**	CO3
4.	Suppose you have a file containing the following characters with the corresponding frequency. A: 45, B: 13, C: 12, D: 16, E: 9, F: 5 If you encode the file with the following codewords identify the size of your file in terms of required bits? A: 0000, B: 0001, C: 0010, D: 0011, E: 0100, F: 0101 Now it is possible to encode the file with fewer number of bits. Build the solution. Develop the solution step by step and finally construct the new code words for each character.		**[7]**	CO4