SEAT NO. 153

# NED UNIVERSITY OF ENGINEERING & TECHNOLOGY

## SECOND YEAR FALL SEMESTER (ELECTRICAL ENGINEERING)
## EXAMINATIONS 2019
## BATCH 2018

**Dated:03-02-2020**

**Time: 3 Hours** **Max.Marks:60**

## Data Structures and Algorithms - EE-264

**Instructions:**

1. **Attempt all Questions.**
2. **In case, if you find any missing information, assume by yourself and properly mention it.**
3. **All questions carry equal (10) Marks.**

**Question 1:** **[CLO1]**

Perform a detailed analysis of SELECTION SORT algorithm including general expression of running time, best case and worst case running times. Also, express the running time in θ-notation.

Pseudo Code:

```
SELECTION SORT
For i = 1 to n:
        min_index = i
        For j = i+1 to n:
            If A[min_index] > a[j]:
                min_index = j
        Swap (A[i].A[min_index])
```

**Question 2:** **[CLO1]**

Perform the time complexity analysis under worst case for Merge Sort Algorithm (based on Divide and Conquer approach). Also, express your answer in relevant asymptotic notation.

Pseudo Code:

```
MERGESORT(A):
n = length(A)
if n < 2:
    return
mid = n/2
left = array of size (mid)
right = array of size (n-mid)
for i = 0 to mid-1:
    left[i] = A[i]
for j = mid to n-1:
    right[j-mid] = A[j]
MERGESORT(left)
MERGESORT(right)
MERGE(left. right, A)
```

```
MERGE(L,R,A):
nL = length(L)
nR = length(R)
i = j = k = 0
while i < nL and j < nR:
        if L[i] < R[j]:
            A[k] = L[i]
            i = i + 1
        else:
            A[k] = R[j]
            j = j + 1
        k = k + 1
while i < nL:
        A[k] = L[i]
        i = i + 1
        k = k + 1
while j < nR:
        A[k] = R[j]
        j = j + 1
        k = k + 1
```

**Question 3:** [CLO1]

Explain the following Asymptotic Notations by considering an example,

1. O Notation
2. $\Omega$ Notation
3. $\Theta$ Notation

**Question 4:** [CLO2]

Demonstrate the mechanism of maintaining data in *stacks* and *queues*. What are their typical operations and policy? Write a python class named *Queue* that uses list structure to store the data inside of it. Also define all the functions, which Queue data structure typically offers, in *Queue* class.

**Question 5:** [CLO2]

You have been asked to develop a system which maintains data of employees in an organization. What would be your preference at implementation level (Traditional Procedural-Oriented programming -TPOP or Object-Oriented programming - OOP)? Take the fundamental components of OOP such as Class, use pseudo-code or C++ or Python and implement some of the following basic tasks related to this application.

a) Create an *Employee* class to store basic information of an employee like name, pay and job.

b) Write a method in *Employee* class to increase the salary of a person with desired percentage.

c) Create a sub-class of *Employee* named *Manager* which replaces the inherited method to increase the salary of a person by including an additional bonus of 10% added in the percentage input of a function.

**Question 6:** [CLO2]

Describe the following functions applied on Dynamic Sets:

1)SEARCH $(S . k)$<br>
2) INSERT $(S . x)$<br>
3) DELETE $(S . x)$<br>
5) MAXIMUM $(S)$<br>
6) PREDECESSOR $(S . x)$

———x———