BÀI TẬP THỰC HÀNH CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

LAB MANUAL

Academic Year : **2022 - 2023**

Semester : **I**

Prepared by

Ms. KhangVQH

S. No.	Experiment
1	**SEARCHING TECHNIQUES**
2	**SORTING TECHNIQUES**
3	**SORTING TECHNIQUES**
4	**IMPLEMENTATION OF STACK AND QUEUE**
5	**APPLICATIONS OF STACK**
6	**IMPLEMENTATION OF SINGLE LINKED LIST**
7	**IMPLEMENTATION OF DOUBLE LINKED LIST**
8	**IMPLEMENTATION OF STACK USING LINKED LIST**
9	**IMPLEMENTATION OF QUEUE USING LINKED LIST**
10	**IMPLEMENTATION OF BINARY SEARCH TREE**

WEEK- 3

SORTING TECHNIQUES

3.1 OBJECTIVE:

1. Sorting the list of integers in ascending order using Selection sort
2. Sorting the list of integers in ascending order using Quick sort

3.2 PROGRAM LOGIC:

Selection sort

1. Read the elements to be sort
2. Select the minimum element
3. Apply the selection sort to sort the remaining elements

Quick sort

1. Read the elements to be sort
2. Find the proper pivot element
3. Apply quick sort method to sort the remaining elements

3.3 IMPLEMENTATION:

- **Selection Sort Algorithm**

```
Algorithm selectionSort ( low, high )
{ //a[low : high] is an array of size n
        i=0, j=0, temp=0, ;
        for i: =low to high do
        {
                minindex = i;
                for  j: =i+1 to  high do
                {
                        if( a[j] < a[minindex] ) then
                                minindex := j;
                }
                temp := a[i];
                a[i] := a[minindex];
                a[minindex] := temp;
        }
}
```

```
void quicksort(int x[10],int first,int last)
{
        int pivot,i,j,t;
        if(first<last)
        {
                pivot=first;
                  i=first;
                  j=last;
```

```
                while(i<j)
                {
                        while(x[i]<=x[pivot]&&i<last)
                                i++;
                         while(x[j]>x[pivot])
                                j--;
                        if(i<j)
                        {
                                t=x[i];
                                x[i]=x[j];
                                x[j]=t;
                        }
                }
                t=x[pivot];
                x[pivot]=x[j];
                x[j]=t;
                quicksort(x,first,j-1);
                quicksort(x,j+1,last);
        }
}
```

3.4 LAB ASSIGNMENT:

1. Apply the selection sort on the following elements 21,11,5,78,49, 54,72,88
2. Rearrange the following numbers using Quick sort procedure. 42, 12, 18, 98, 67, 83, 8, 10, 71

3.5 POST-LAB QUESTIONS:

1. What is the time complexity of selection sort
2. What is the time complexity of quick sort
3. Why sorting is required
4. Is selection sort is stable
5. What is the worst case for quick sort