

Experiment No 4:

A program for Selection Sort

Aim: Write a program to implement Selection Sort.

Theory:

This sorting algorithm is an in-place comparison-based algorithm in which the list is divided into two parts, the sorted part at the left end and the unsorted part at the right end. Initially, the sorted part is empty and the unsorted part is the entire list.

The smallest element is selected from the unsorted array and swapped with the leftmost element, and that element becomes a part of the sorted array. This process continues moving unsorted array boundary by one element to the right.

***Any Solved Example with all passes**

Algorithm:

Algorithm Selection sort:

Procedure 1

MIN(A,K,N,LOC)

An array A is in memory. This procedure finds the location LOC of the smallest element among A[K], A[K+1], ..., A[N].

1. Set MIN:=A[k] and LOC:=K. [Initializes pointers]

2. Repeat for J=K+1, K+2, N:

If MIN>A[J], then: set MIN:=A[j] and LOC:=A[j] and LOC:=J.

[End of loop.]

3. Return

Algorithm:

(Selection Sort) SELECTION(A,N)

This algorithm sorts the array A with N elements.

1. Repeat Steps 2 and 3 for K=1, 2, ..., N-1:

2. Call MIN(A,K,N,LOC).

3. [Interchange A[K] and A[LOC].]

Set TEMP:=A[K], A[K]:=A[LOC] and A[LOC]:=TEMP.

[End of Step 1 loop.]

4. Exit

PROGRAM: [Write program for selection sort-out put should be display in passes]

```
#include<stdio.h>
```

```

void main()
{
    int n,k,i,temp,loc;
    printf("Enter number of the elements : ");
    scanf("%d",&n);
    int arr[n];
    for (i=1;i<=n;i++)
    {
        printf("Enter %d element :",i);
        scanf("%d",&arr[i]);
    }

    for(k=1;k<=n;k++)
    {
        loc = min(arr,k,n);
        temp = arr[loc];
        arr[loc] = arr[k];
        arr[k] = temp;
        for (i=1;i<=n;i++)
        {
            printf("%d , ",arr[i]);

        }
        printf("\n");
    }
    printf("Sorted Array : \n");
    for (i=1;i<=n;i++)
    {
        printf("%d ,",arr[i]);

    }
}

int min(int *arr,int k,int n)
{
    int min = k;
    for(min=k;k<=n;k++)
    {
        if(arr[min]>arr[k])
        {
            min = k;
        }
    }
    return min;
}

```

OUTPUT

CONCLUSION: