

Practical no -2

Aim: Write a program to sort the array element using selection sort.

Selection sort:

this simplest method of sorting. the method is to arrange the element in ascending or descending order. for to arrange the element in ascending order ,to begin with the 0th element is compared with the all remaining element .if it is found to be greater than the all the element then they are interchanged. So after the first iteration smallest element gets placed at the first location or first position. Similarly, in the second iteration the second lowest element get placed at the second se position in the list. As a result after all the iteration the list becomes a sorted list.

Analysis

Each of these algorithms requires $n-1$ passes: each pass places one item in its correct place. (The n^{th} is then in the correct place also.) The i^{th} pass makes either i or $n - i$ comparisons and moves. So:

$$\begin{aligned} T(n) &= 1 + 2 + 3 + \dots + (n - 1) \\ &= \sum_{i=1}^{n-1} i \\ &= \frac{n}{2}(n - 1) \end{aligned}$$

or $O(n^2)$ - but we already know we can use heaps to get an $O(n \log n)$ algorithm. Thus these algorithms are only suitable for small problems where their simple code makes them faster than the more complex code of the $O(n \log n)$ algorithm. As a rule of thumb, expect to find an $O(n \log n)$ algorithm faster for $n > 10$ - but the exact value depends very much on individual machines!.

Algorithm:

Selection_sort (array,size)

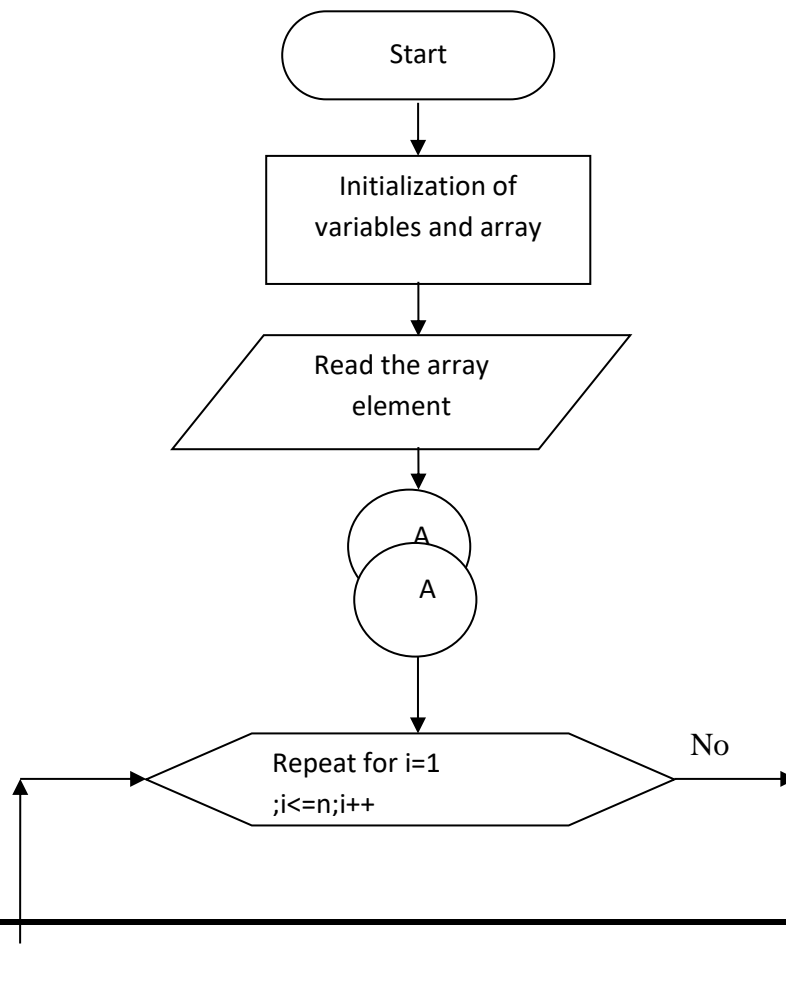
Where array->represents the list of element.

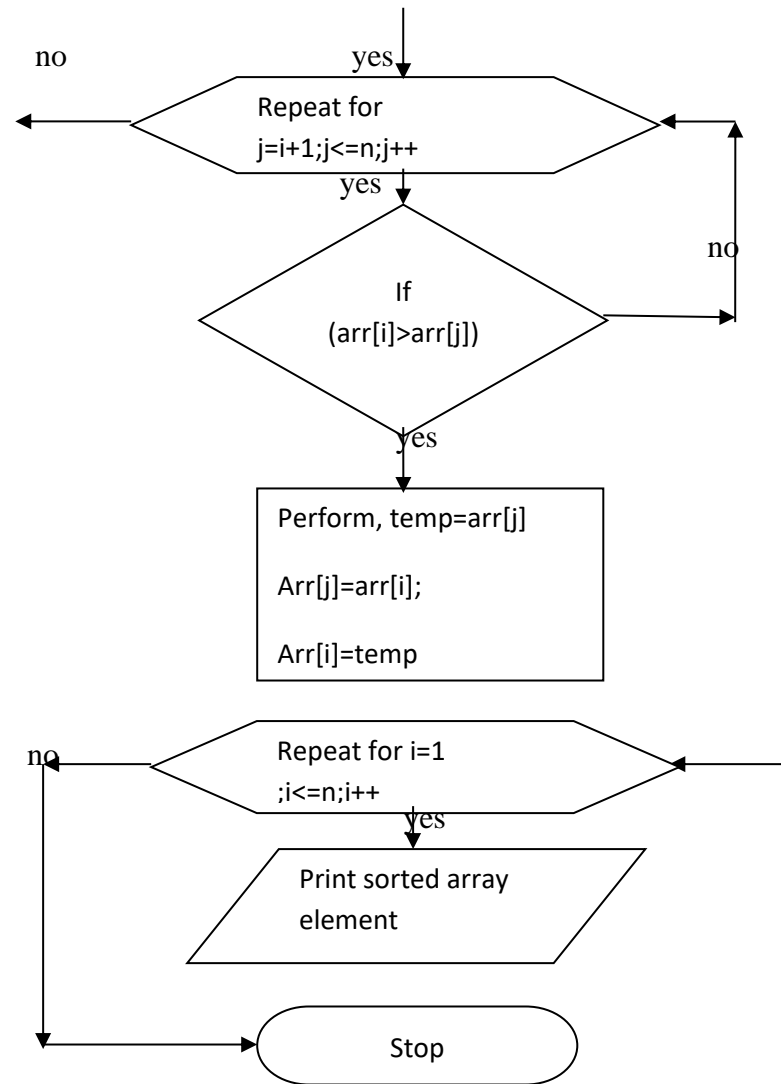
Size-> represents the size of .

Steps:

1. Current= 0 [initially]
2. Repeat through step 7 while (current<size)
3. j=current+1
4. Repeat through step 6 while (j<size)
5. if(array[current]>array[j])
 - a)temp=array[current];
 - b)array [current] =array[j];
 - c)array[j]=temp;
6. j=j+1;
7. current=current+1;
8. Exit

Flowchart:





Program:

```
#include<conio.h>
```

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

```
void main()
```

```
{
```

```
int arr[50],i,j,temp,dim;
```

```
clrscr();
```

```
printf("Enter the dimension");
```

```
scanf("%d",&dim);
```

```
printf("\nEnter the array element:\n");
```

```

for(i=1;i<=dim;i++)
    {
        scanf("%d",&arr[i]);
    }
for(i=1;i<=dim;i++)
{
    for(j=i+1;j<=dim;j++)
    {
        if(arr[i]>arr[j])
        {
            temp=arr[i];
            arr[i]=arr[j];
            arr[j]=temp;
        }
    }
}

printf("\nSorted array is:-\n");
for(i=1;i<=dim;i++)
    printf("%d ",arr[i]);

getch();
}

```

Output:

Enter the dimension 5

Enter the array element:

8

9

5

4

3

Sorted array is:-

3

4

5

8

9

