Experiment No. 5

Aim: Implementation of selection sorting technique considering a real world application.

Objective: To impart knowledge of sorting and searching algorithm.

Theory:

1) Introduction to sorting:
Sorting is the process of arranging the elements of an array so that they can be placed either in of an array so that they can be placed either ascending or descending order for eg consider any array $A = \{A_1, A_2, A_3, A_4 \ldots A_n\}$ the array is called to be in an ascending order if element of A are arranged like $A1 > A2 > A3 > A4 > \ldots > An$

2) Types of sorting:
① Bubble sort:-
It is the simplest sort methods which performs sorting by repeatedly moving the largest element to the highest index of array It comprises of comprising each element to its adjacent element and replace them accordingly

② Insertion sort:
The insertion sort inserts each element of the array to its proper place. It is very simple sort method which is used to arrange the deck of cards while playing triedge.

(iii) Selection sort :

selection sort finds the smallest element in the array and place it on the first place of the list, then array and place it on the second smallest element in the array & place. This process continues until all elements are moved to their correct order.

(iv) Merge sort :

Merge sort follows divide and conquer approach in which, the list is first divided into the sets of equal elements and then each half of the list is sorted by using merge sort.

3) Introduction to selection sort :

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

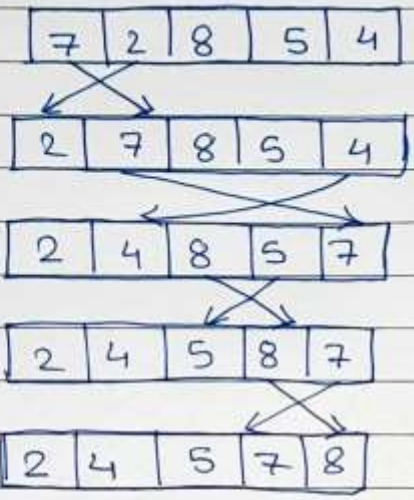
4) Algorithm :

selection sort [A[0... n-1])
// sorts a given array by selection sort
// input : An array A[0.... n-1] of orderable elements
// output : Array A[0... n-1] sorted in ascending order

For i ← 0 to n-2 do
min ← i

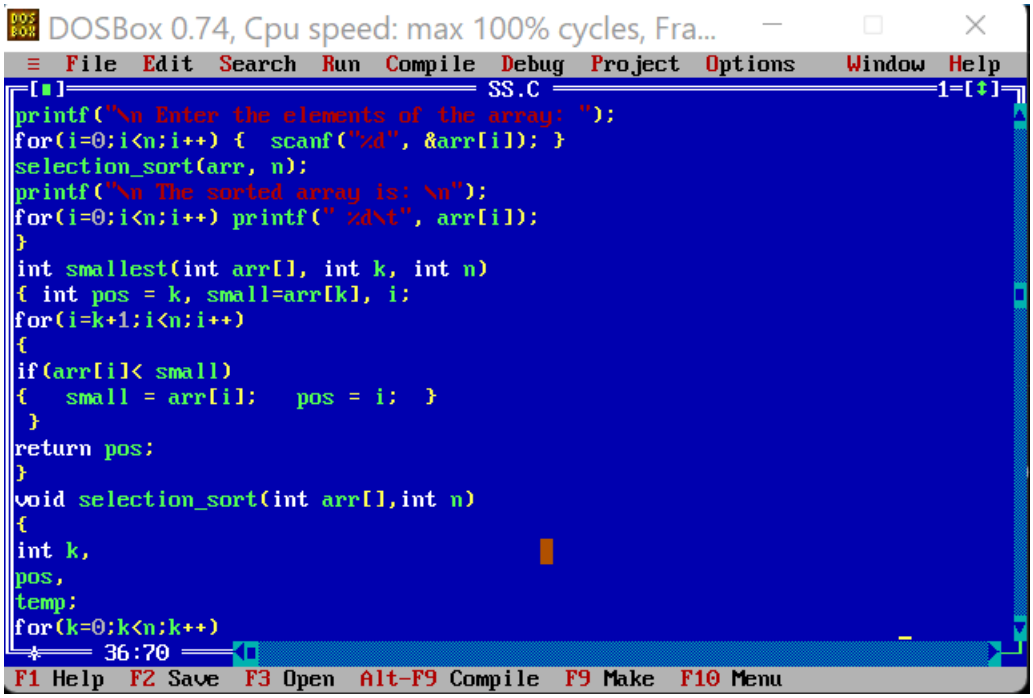
for j ← i+1 to n-1 do
if A[j] < A[min] min ← j
swap A[i] and A[min]

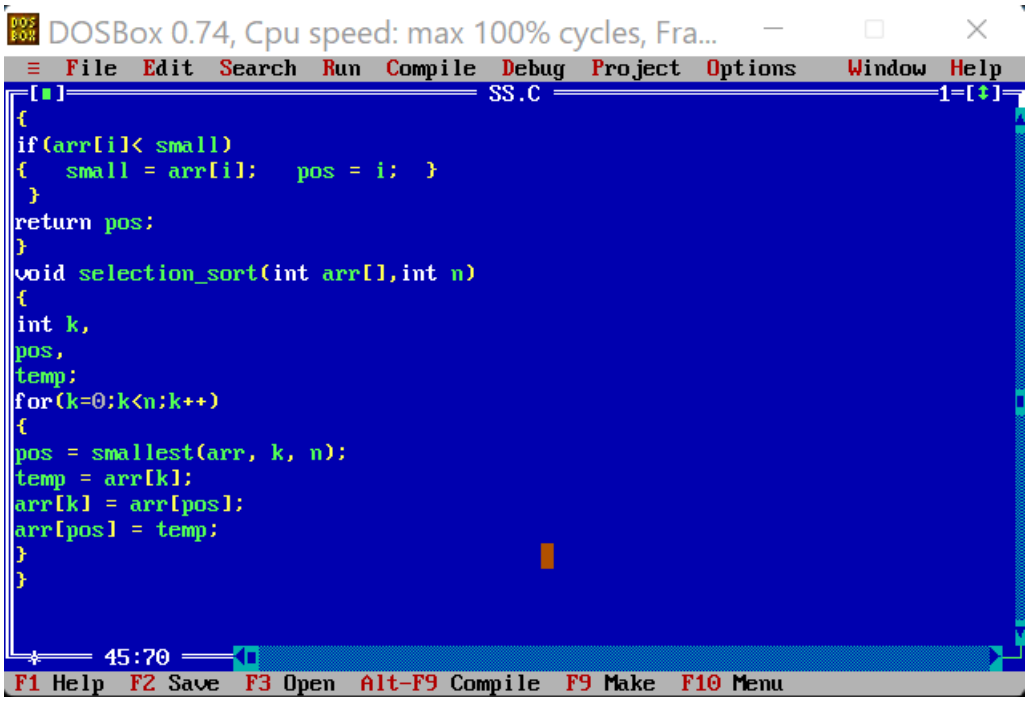
5) Example :

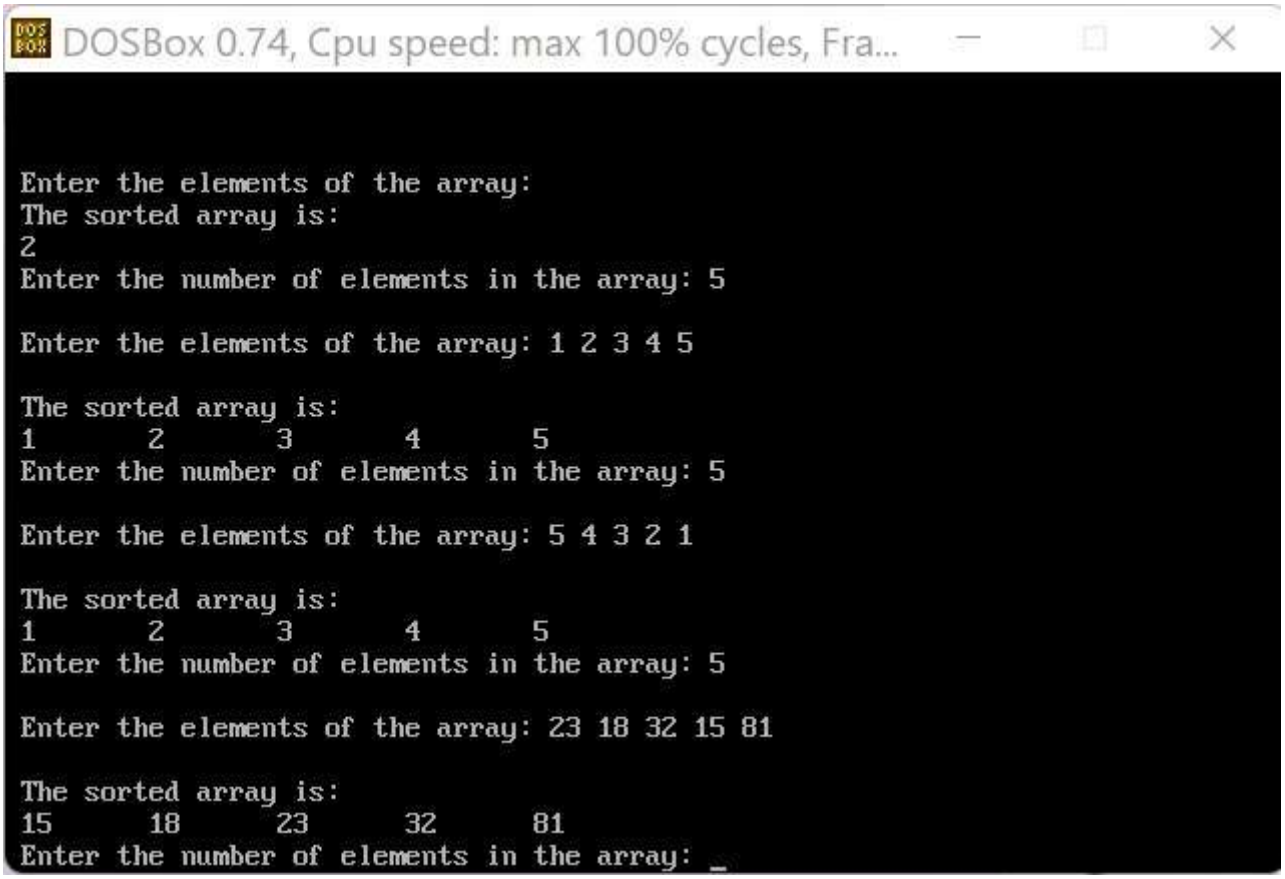


conclusion : Selection sort is sorting algorithm known by its simplicity. Unfortunately it lacks efficiency on huge lists of items and also it does not stop unless the number of iterations has been achieved even though the list is already sorted.

Outcome :
Implement sorting and searching techniques for real world applications.


DOSBox 0.74, Cpu speed: max 100% cycles, Fra...
File Edit Search Run Compile Debug Project Options Window Help
SS.C
printf("\n Enter the elements of the array: ");
for(i=0;i<n;i++) { scanf("%d", &arr[i]); }
selection_sort(arr, n);
printf("\n The sorted array is: \n");
for(i=0;i<n;i++) printf(" %d\t", arr[i]);
}
int smallest(int arr[], int k, int n)
{ int pos = k, small=arr[k], i;
for(i=k+1;i<n;i++)
{
if(arr[i]< small)
{ small = arr[i]; pos = i; }
}
return pos;
}
void selection_sort(int arr[],int n)
{
int k,
pos,
temp;
for(k=0;k<n;k++)
36:70
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


DOSBox 0.74, Cpu speed: max 100% cycles, Fra...
File Edit Search Run Compile Debug Project Options Window Help
SS.C
{
if(arr[i]< small)
{ small = arr[i]; pos = i; }
}
return pos;
}
void selection_sort(int arr[],int n)
{
int k,
pos,
temp;
for(k=0;k<n;k++)
{
pos = smallest(arr, k, n);
temp = arr[k];
arr[k] = arr[pos];
arr[pos] = temp;
}
}
45:70
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


DOSBox 0.74, Cpu speed: max 100% cycles, Fra...
Enter the elements of the array:
The sorted array is:
2
Enter the number of elements in the array: 5
Enter the elements of the array: 1 2 3 4 5
The sorted array is:
1 2 3 4 5
Enter the number of elements in the array: 5
Enter the elements of the array: 5 4 3 2 1
The sorted array is:
1 2 3 4 5
Enter the number of elements in the array: 5
Enter the elements of the array: 23 18 32 15 81
The sorted array is:
15 18 23 32 81
Enter the number of elements in the array: _