Hello Everyone

Course Code : CSE201

Course Title : Data Structure

Course Teacher:

Md. Jamal Uddin

Assistant Professor

Department of CSE,BSMRSTU,

Gopalganj-8100.

Presented By:

Tushar Sarkar

Student ID: 18CSE035

Department of CSE,BSMRSTU,

Gopalganj-8100.

Selection Sort

Outline

- Introduction to Sort
- What is Selection Sort?
- Algorithm Explain
- Example of Selection Sort
- Pseudo Code
- Complexity
- Advantages & Disadvantages
- Conclusion

Introduction to Sort

- **Sorting** is one of the most basic functions applied to data.
- Sorting is a technique that is implemented to arrange the data in a specific order.
- It means arranging the data in a particular fashion, which can be increasing or decreasing.
- Sorting is required to ensure that the data which we use is in a particular order so that we can easily retrieve the required piece of information from the pile of data.
- If the data is unkempt and unsorted, when we want a particular piece of information, then we will have to search it one by one every time to retrieve the data.

What is Selection Sort?

- **Selection sort** is a simple **sorting** algorithm. ...
- 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.
- The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array

Algorithm Explain

Step 1 − Set MIN to location 0

Step 2 − Search the minimum element in the list

Step 3 − Swap with value at location MIN

Step 4 − Increment MIN to point to next element **S**

Step 5 − Repeat until list is sorted

Example of Selection Sort

If primary array is :

9	13	15	11	2

Sorting By Ascending Order :

2	9	11	13	15

Sorting By descending Order :

15	13	11	9	2

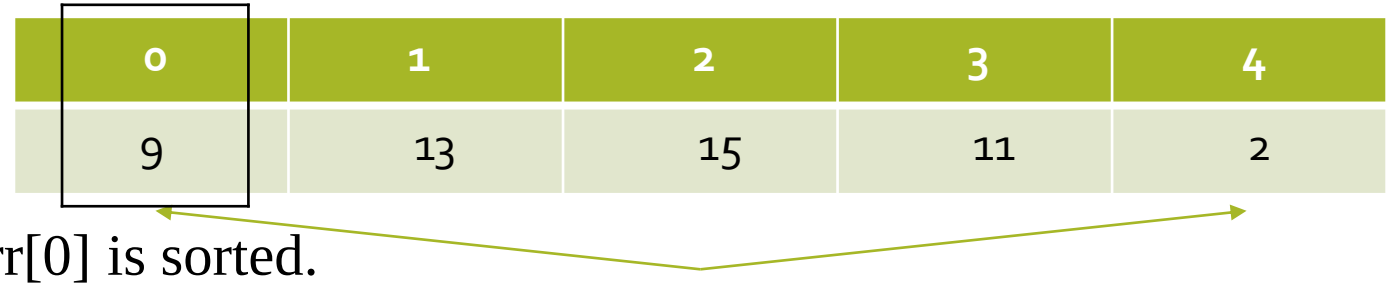
Example Explain

Iteration 1:

In terms of index 0

arr[0]=9,arr[4]=2

arr[0]>arr[4], swap(arr[0],arr[4]) & arr[0] is sorted.

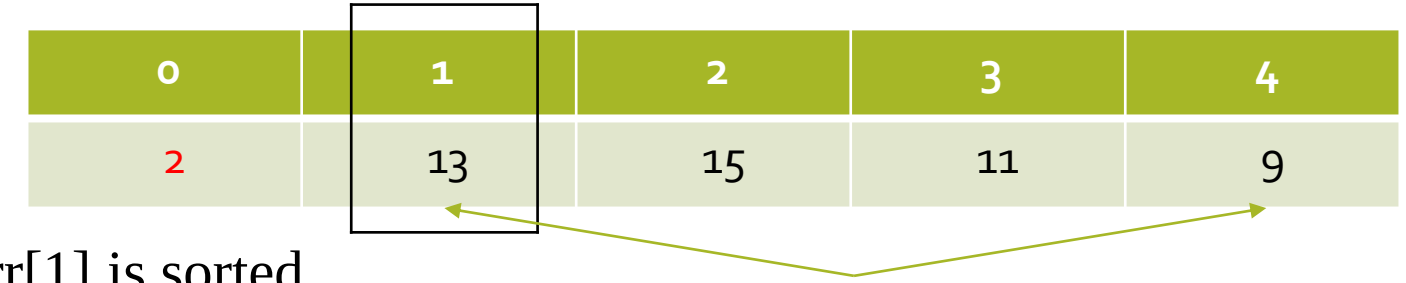


Iteration 2:

In terms of index 1

arr[1]=13,arr[4]=9

arr[1]>arr[4], swap(arr[1],arr[4]) & arr[1] is sorted.



Example Explain

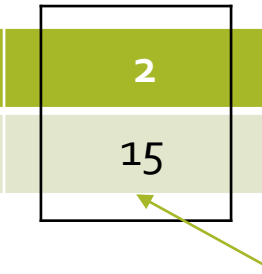
Iteration 3:

In terms of index 2

arr[2]=9,arr[3]=2

arr[2]>arr[3], swap(arr[2],arr[3]) & arr[2] is sorted.

Index	0	1	2	3	4
Value	2	9	15	11	13



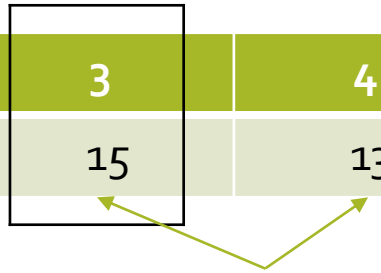
Iteration 4:

In terms of index 3

arr[3]=15,arr[4]=13

arr[3]>arr[4], swap(arr[3],arr[4]) & arr[3] is sorted.

Index	0	1	2	3	4
Value	2	9	11	15	13



Example Explain

Iteration 5:

In terms of index 4

Here present iterate is last index

So no iterate & already array is sorted.

Index	0	1	2	3	4
Value	2	9	11	13	15

Finally Sorted array is :

Index	0	1	2	3	4
Value	2	9	11	13	15

Pseudo Code

D_Shortest_Path_Queri | A_Reconnaissance.cpp | Linear_Search.cpp | Quick_Sort.cpp | Selection_Sort.cpp | B_Borze.cpp | C_Fair_Candy_Distribution.cpp | 11631_Dark_roads.cpp | 10048_A

```cpp
int Selection_Sort(int A[],int N){
        int i,j,temp,min_index;
    for(i = 0; i < N-1; i++){
        min_index = i;
        for(j = i+1; j < N; j++){
            if(A[j] < A[min_index]){
                min_index = j;
            }
        }
        if(i != min_index){
            temp = A[min_index];
            A[min_index] = A[i];
            A[i] = temp;
        }
    }
}
//Main Funcion_Tushar_______
int main()
{
    int N,i;
    cout << "Enter Array Size : ";
    cin >> N;

    int A[N+1];

    cout << "The elements of Array : ";
    for(i = 0; i < N; i++)cin >> A[i];

    Selection_Sort(A,N);

    cout << "After Sorting Array   : ";
    for(i = 0; i < N; i++){
        cout << A[i] << " ";
    }
```

```
C:\WINDOWS\system32\cmd.exe - pause
Enter Array Size : 5
The elements of Array : 9 13 15 11 2
After Sorting Array   : 2 9 11 13 15
Press any key to continue . . .
```

Complexity

- Selecting the minimum requires scanning n elements & then swapping it into the first position.

$$(n-1)+(n-2)+\ldots.+1=\sum_{i=1}^{n-1} i$$

- By Arithmetic Progression:

$$\sum_{i=1}^{n-1} i = \frac{(n-1)+1}{2}(n-1)=\frac{1}{2}n(n-1)=\frac{1}{2}(n^2-n)$$

Which is of complexity $O(n^2)$ it terms of number of comparisons.

Advantages of Selection Sort

- Easy to implement;
- Requires no additional storage space.
- It performs well on a small list.

Disadvantages of Selection Sort

- Poor efficiency when dealing with a huge list of items
- Its performance is easily influenced by the initial ordering of the items before the sorting process.

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 (n-1, n is the number of elements) even though the list is already sorted.

Thank You