9/16/25, 9:51 PM NamasteDev

Interview Practice Ne

Courses

Join Community





Selection Sort

Selection Sort

Selection Sort is a simple comparison-based sorting algorithm.

It divides the array into two parts: a **sorted subarray** and an **unsorted subarray**.

Initially, the **sorted part is empty**, and the unsorted part is the entire array.

In each iteration, it finds the minimum element from the unsorted part and moves it to the end of the sorted part.

Example 1:

Input: [4, 5, 1, 3, 9]

Output: [1, 3, 4, 5, 9]

Approach:

Iterate over the array from index 0 to n-2.

For each index i, assume the element at i is the minimum in the unsorted part.

Run an inner loop from j = i+1 to n-1 to find the actual minimum element.

If a smaller element is found, update the $\,\,\mbox{min}\,\,$ index .

After the **inner loop, swap the element** at i with the element at min (if they're not the same).

Repeat until the array is sorted.

Time & Space Complexity:

Time Complexity: $O(n^2)$ In all cases best, average and worst.

Roughly n*(n-1)/2 comparisons are always performed.

9/16/25, 9:51 PM NamasteDev

Space Complexity: O(1) Selection Sort is an in-place sorting algorithm, so it doesn't require extra space.

Dry Run

```
Input: arr = [4, 5, 1, 3, 9] 

i = 0 \rightarrow \min_i dx = 0 \ j = 1 \rightarrow arr[1] = 5 \Rightarrow arr[0] = 4 \rightarrow no \ change \ j = 2

\rightarrow arr[2] = 1 < arr[0] = 4 \rightarrow \min_i dx = 2 \ j = 3 \rightarrow arr[3] = 3 \Rightarrow arr[2] = 1 \rightarrow no \ change \ j = 4 \rightarrow arr[4] = 9 \Rightarrow arr[2] = 1 \rightarrow no \ change \ swap \ arr[0] \Rightarrow arr[2] \rightarrow [1, 5, 4, 3, 9] \ i = 1 \rightarrow \min_i dx = 1 \ j = 2 \rightarrow arr[2] = 4 < arr[1] = 5 \rightarrow \min_i dx = 2 \ j = 3 \rightarrow arr[3] = 3 < arr[2] = 4 \rightarrow \min_i dx = 3

j = 4 \rightarrow arr[4] = 9 \Rightarrow arr[3] = 3 \rightarrow no \ change \ swap \ arr[1] \Rightarrow arr[3] \rightarrow [1, 3, 4, 5, 9] \ i = 2 \rightarrow \min_i dx = 2 \ j = 3 \rightarrow arr[3] = 5 \Rightarrow arr[2] = 4 \rightarrow no \ change \ j = 4 \rightarrow arr[4] = 9 \Rightarrow arr[2] = 4 \rightarrow no \ change \ no \ swap \rightarrow [1, 3, 4, 5, 9] \ Final Sorted Array: [1, 3, 4, 5, 9]
```

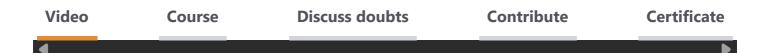
```
JavaScript Python Java C++ C C#
```

```
let arr = [4, 5, 1, 3, 9];

function selectionSort(arr) {
    let n = arr.length;
    for (let i = 0; i < n - 1; i++) {
        let min = i;
        for (let j = i + 1; j < n; j++) {
            if (arr[j] < arr[min]) {
                min = j;
            }
        }
        if (min != i) {
            let temp = arr[i];
            arr[min];
        arr[min] = temp;
        }
}</pre>
```

9/16/25, 9:51 PM NamasteDev

```
}
}
return arr;
}
let result = selectionSort(arr);
console.log("Sorted array", result);
```



Selection Sort - DSA Notes

Selection Sort - DSA Notes