Sorting Techniques

Sorting - I

Date
Page

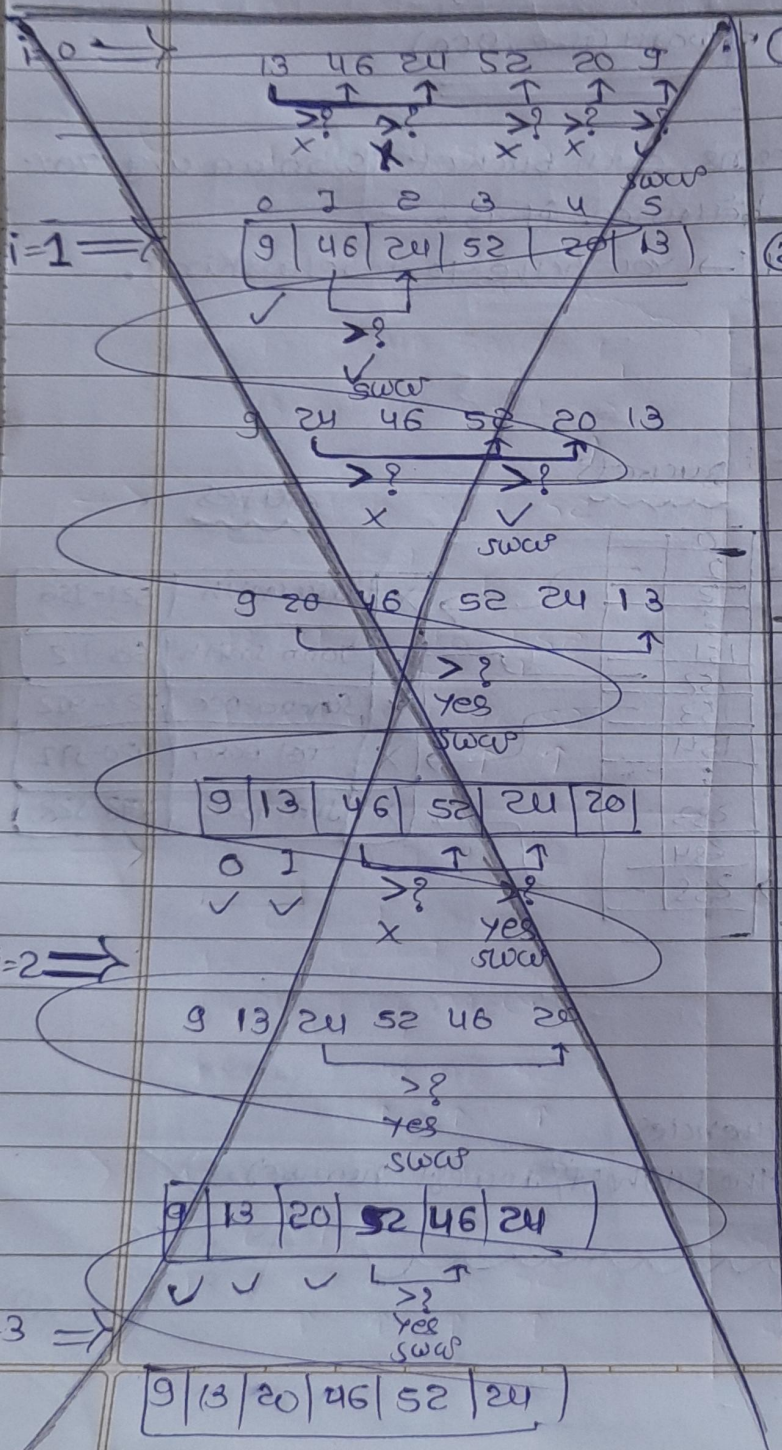
Selection Sort

13	46	24	52	20	9
0	1	2	3	4	5

↑ → check is > ? or not

✓ → yes, swap

* Select minimums
* minimum elements start getting sorted first.



① 13 46 24 52 20 9
i=0 ↑ ↑ ↑ ↑ ↑ ✓
9 46 24 52 20 13
✓

② 9 46 24 52 20 13
i=1 ↑ ✓
9 24 46 52 20 13
i=1 ↑ ↑ ✓
9 20 46 52 24 13
i=1 ↑ ✓
9 13 46 52 24 20
✓ ✓

③ 9 13 46 52 24 20
i=2 ↑ ↑ ✓
9 13 24 52 46 20
i=2 ↑ ✓
9 13 20 52 46 24
✓ ✓ ✓

④ 9 13 20 52 46 24
i=3 ↑ ✓
9 13 20 46 52 24
i=3 ↑ ✓
9 13 20 24 52 46
✓ ✓ ✓ ✓

⑤ 9 13 20 24 52 46
i=4 ↑

9	13	20	24	46	52
✓	✓	✓	✓	✓	✓

```
selection-sort (arr, size)
{   for (i=0 to size)
    {  for (j=i to size)
       {   if (arr[i] > arr[j])
           {  swap(arr[i], arr[j]); }
       }
    }
}
```

=> Here swapping happens multiple times.

Efficient code :-

```
selection-sort (arr, size)
{   for (i=0 to size)
    {   int min = i;        // min index
        for (j=i+1 to size)
        {   if (arr[min] > arr[j])
            {  min = j; }
        }
        swap(arr[i], arr[min]);
    }
}
```

TC :- $O(n^2)$

SC :- $O(1)$

Bubble Sort

13	46	24	52	20	9
0	1	2	3	4	5

* opposite of selection sort
* maximum elements start getting sorted first (Bubbles up)

TC :- $O(n^2)$
SC :- $O(1)$

=> 13 46 24 52 20 9

13 24 46 52 20 9

13 24 46 52 20 9

13 24 46 20 9 52

=> 13 24 46 20 9 52

13 24 20 46 9 52

13 24 20 9 46 52

=> 13 24 20 9 46 52

13 20 24 9 46 52

13 20 9 24 46 52

=> 13 20 9 24 46 52

13 9 20 24 46 52

=> 13 9 20 24 46 52

9	13	20	24	46	52

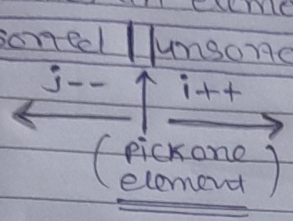
sorted

```
Bubble_sort (arr, size)
for (int i = 0 to size-1)
{   Bool swapped = false;
    for (int j = 0 to size-i-1)
    {   if (arr[j] > arr[j+1])
        {   swap (arr[j], arr[j+1]); }
            swapped = true;
    }
    if (!swapped) break;
}
```

swapped boolean variable to check if array is already sorted.

Insertion Sort

* Takes an element & sorts it.
* sorted | unsorted array



=> | 13 | 46 24 52 20 9 |

sorted ↑ unsorted

key

Inner working

Insert value of key

| 13 | 46 | 46 | 52 | 20 | 9 |

13 | 24 | 46 | 52 | 20 | 9

| 13 | 46 | 24 52 20 9 |

sorted | unsorted

key = 24

| 13 | 24 | 46 | 52 20 9 |

sorted | unsorted

key

| 13 | 24 | 46 | 52 | 20 9 |

sorted | unsorted

0 1 2 3 4 key 5

| 13 | 20 | 24 | 46 | 52 | 9 |

sorted ↑ key

Inner working (arr[j+1] = arr[j])

j=4 13 20 24 46 52 52

j=3 13 20 24 46 46 52

j=2 13 20 24 24 46 52

j=1 13 20 20 24 46 52

j=0 13 13 20 24 46 52

| 9 | 13 | 20 | 24 | 46 | 52 |

sorted

Insert value of key arr[j+1] = key

j=-1 9 13 20 24 46 52

```
for(i=1; i<size; i++)
{   key = arr[i];
    for(j=i-1; j>=0 && key<arr[j]; j--)
    {   arr[j+1] = arr[j]; }
    arr[j+1] = key;
}
```

TC :- O(n²) / SC :- O(1)