Bubble Sort :

19 July 2023 14:07

- works opposite to selation sort. push the max element to the last by
- adjacent swaps
- after each iteration, max element is bubbled to the last

working

0 ← i

5	10	2	0	9
0	1	2	3	4

compare if sorted : sorted ignore

5	10	2	0	9

compare if sorted : swap (2,10)

5	2	10	0	9

compare if sorted : swap (0, 10)

5	2	0	10	9

compare if sorted swap (9, 10)

5	2	0	9	10

→ sorted part

- iteration 1 : Done .

- Iteration 2:

0 ← i

5	2	0	9	10

5	2	0	9	10

compare if sorted : swap (2, 5)

Dry Run

19 July 2023 14:20

2	5	0	9	10

compare if sorted : swap(0, 5)

2	0	5	9	10

compare if sorted : ignore

2	0	5	9	10

↳ sorted part

- Iteration 3:

0 ← i

2	0	5	9	10

compare if sorted : swap (0,2)

0	2	5	9	10

compare if sorted : ignore

0	2	5	9	10

↳ sorted part

- Iteration 4:

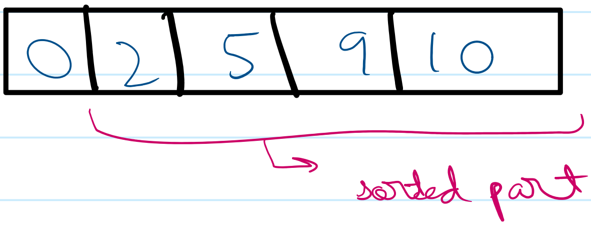
0 ← i

0	2	5	9	10

compare if sorted : ignore

Algo and Pseudocode

19 July 2023 14:26



0	2	5	9	10

sorted part.

- Single element is always sorted.

Algorithm :

1: select the range of unsorted array, using an outer loop (starting from end). loop will backward

2: run an inner loop from 0 → i-1. pushing the maximum element to the last by swapping.

3: make adjacent swaps by (a[j] > a[j+1]).

4: after each iteration, last part will become sorted

after (n-1) iteration, array will be sorted.

Pseudo Code:

```
for(i=n-1; i>=0){
    for(j=0; j<=i-1){
      if(arr[j] >= arr[j+1])
        swap();
    }
}
```

CODE.cpp

19 July 2023 14:46

C++ :

```
void bubbleSort(vector<int> &arr, int n)
{ // outer loop to iterate over the unsorted part
  for (int i = n - 1; i >= 0; i--)
  {
    // inner loop to push the maximum element to the last by
ajacent swaps
    for (int j = 0; j <= i - 1; j++)
    {
      if (arr[j] > arr[j + 1])
        swap(arr[j], arr[j + 1]);
    }
  }
}
```

Time Complexity : $O(n^2)$

Space Complexity : $O(1)$