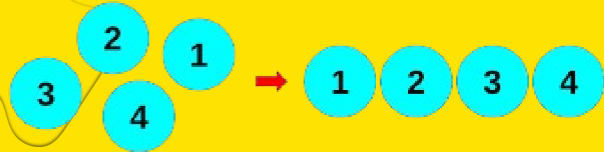


+++++
+++++
+++++
+++++

Sorting algorithm



Understanding Bubble Sort-1



>>>>>

@KRISHAN KUMAR

|| follow ||
for more update

Sorting Algorithm

Sorting →

When we have data and we arrange that data, where we bring the meaning of that data, that called → Sorting

↳ Sorting can be ^{arran} easy for human in short data like

[3, 4, 1, 5, 9, 19] sorting → [1, 3, 4, 5, 9, 19]

↳ But where Huge data like → in Millions for sorting we need computer. by this we can solved this problem using python programming.



Types of Sorting algorithm →

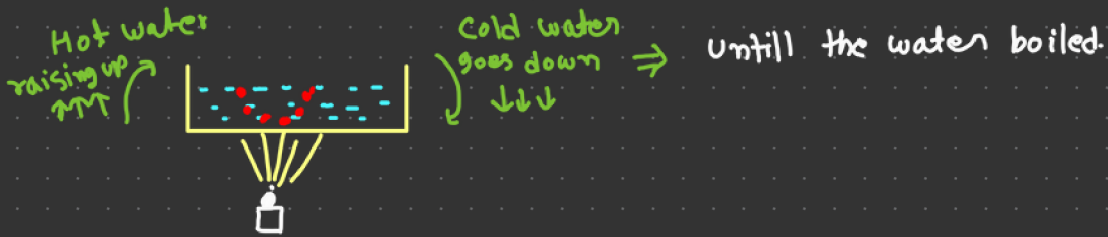
- Bubble sort algorithm
- Selection sort algorithm
- Insertion sort algorithm



Bubble sort algorithm



inspire by Bubble → when water heated



the same step. follow in the bubble sort algorithm

Algorithm

[5, 3, 1, 2, 4] Apply bubble sort on this list

Step 1:-

$\xrightarrow{\quad}$
 [5, 3, 1, 2] ⇒ Start moving towards right one by one and if we find a number which is bigger, we have to push him / Bubble him to the right.

Step 2:-

$\xrightarrow{\text{comparing}}$
 [5, 3, 1, 2, 4]

Here when we comparing 5, 3 and i show that 5 is bigger than 3, so we swap 5 to the right. And the

list is → [3, 5, 1, 2, 4] → Again go and compare (5, 1)

[3, 1, 5, 2, 4] → Again compare and swap

[3, 1, 2, 5, 4] → compare & swap

[3, 1, 2, 4, 5] → Now Repeat for the Next (3, 1, 2, 4) element.

Rule:-

① $\text{range}(5) \rightarrow 0, 1, 2, 3, 4$

② $\text{range}(1, 5) \rightarrow 1, 2, 3, 4$

③ $\text{range}(5, 0, -1) \rightarrow 5, 4, 3, 2, 1$ (-ve direction)

```
# bubble sorting
def bubbleSort(arr):
    Round
    ① ② ③
    3/2/1
    for i in range(len(arr)-1, 0, -1):
        for j in range(i):
            if (arr[j] > arr[j+1]):
                arr[j], arr[j+1] = arr[j+1], arr[j]  # Swapping
```

Let $\rightarrow \text{arr} \rightarrow [4, 2, 3, 1]$, $\text{len}(\text{arr}) \rightarrow 4$

here $\rightarrow \text{len}(\text{arr}) - 1 \rightarrow 4 - 1 \Rightarrow 3$

means $\rightarrow \text{for } i \text{ in range}(\text{len}(\text{arr}) - 1)$, here $\Rightarrow i = 3$

and $\text{for } j \text{ in range}(i)$: , here $j = 0$

Sorting \rightarrow

Step 2 $i = 2$

$[2, 3, 1, 4]$ \rightarrow no swapping

\downarrow

$[3, 2, 1, 4]$

\downarrow

$[2, 1, 3, 4]$ done

Step 3 $i = 1$

$[2, 1, 3, 4]$ \rightarrow no swapping

\downarrow

$[1, 2, 3, 4]$

Step 1 $i = 3$

$\{4, 2, 3, 1\}$

\downarrow Swapping

$[2, 4, 3, 1]$

\downarrow

$[2, 3, 4, 1]$

\downarrow

$[2, 3, 1, 4]$

Bubble Sorting done

{ Swapping for the 1st element
done \rightarrow going 4th outer loop }

0.2 Rounds of i taken 1 by 1

```
]: # bubble sorting
```

```
def bubbleSort(arr):
```

```
    for i in range(len(arr)-1, 0, -1):
```

```
        for j in range(i):
```

```
            if arr[j] > arr[j+1]:
```

```
                arr[j], arr[j+1] = arr[j+1], arr[j]
```

```
]: arr = [7,4,6,5,2]
```

```
    bubbleSort(arr)
```

```
    print(arr)
```

```
[2, 4, 5, 6, 7]
```

How this code work internally



Step: 1

$j=4$
[7, 4, 6, 5, 2]

[4, 7, 6, 5, 2]

[4, 6, 7, 5, 2]

[4, 6, 5, 7, 2]

[4, 6, 5, 2, 7]

internal loop done
go to outer loop
and Repeat the
same process
until all value get
Sorted

Step: 2

$j=3$
[4, 6, 5, 2, 7]

[4, 6, 5, 2, 7]

[4, 5, 6, 2, 7]

[4, 5, 2, 6, 7] done

Step: 3

$j=2$
[4, 5, 2, 6, 7]

[4, 5, 2, 6, 7]

[4, 2, 5, 6, 7] done

Step: 4

$j=1$
[4, 2, 5, 6, 7]

[2, 4, 5, 6, 7]

Bubble Sorting done

Q What is the time complexity of this code.

```
: # bubble sorting  
  
def bubbleSort(arr):  
    for i in range(len(arr)-1, 0, -1): → outer loop  
        for j in range(i):  
            if (arr[j] > arr[j+1]):  
                arr[j], arr[j+1] = arr[j+1], arr[j]  
            ↙ inner loop
```

↳ outer loop runs $\Rightarrow n$ times
inner loop runs $\Rightarrow n$ times

So that time complexity of this code is

$$T(c) = O(n \times n)$$

$$\boxed{T(c) = O(n^2)}$$

Q In the case of worst case How many times comparison will be happen in this code?

↳ In the worst case every time we will doing the comparison, so

the no. of loop is running $\Rightarrow n^2$

So the in worst case comparison happens $\rightarrow n^2$ time

Q What is the worst case in this situations?

↳ When array already is reversing or sorted in descending order. for example $\rightarrow \{5, 4, 3, 2, 1\} \Rightarrow$ Already Sorted (worst case)