Bubble Sort

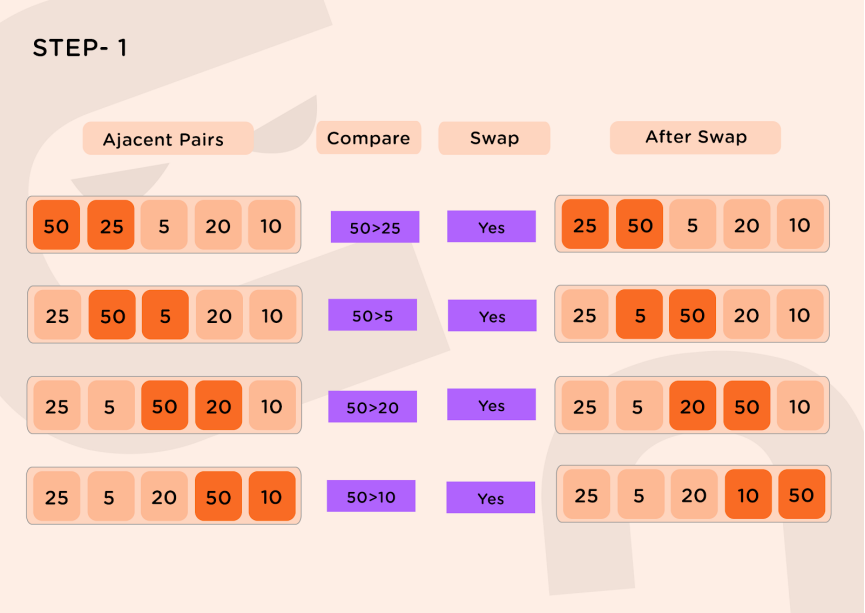
In selection sort, the elements from the start get placed at the correct position first and then the further elements, but in the bubble sort, the elements start to place correctly from the end.

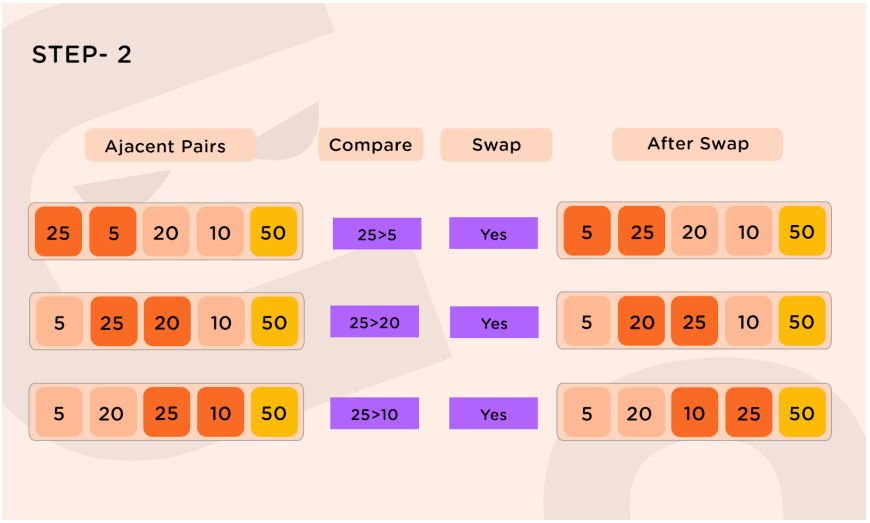
In this technique, we just compare the two adjacent elements of the array and then sort them manually by swapping if not sorted. Similarly, we will compare the next two elements (one from the previous position and the corresponding next) of the array and sort them manually. This way the elements from the last get placed in their correct position. This is the difference between selection sort and bubble sort.

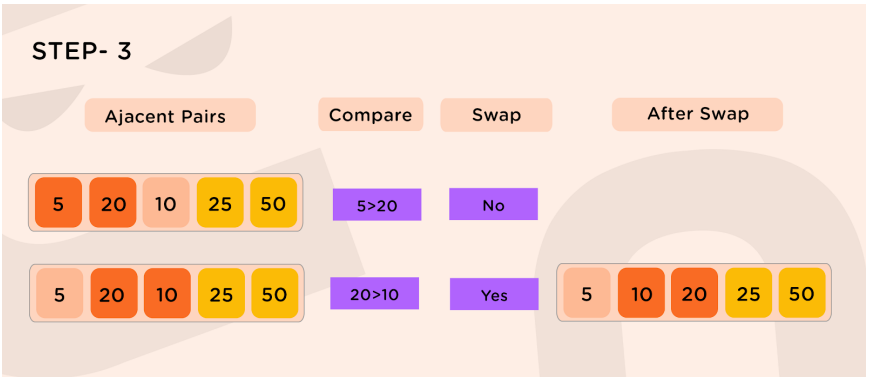
Consider the following depicted array as an example. You want to sort this array in increasing order.

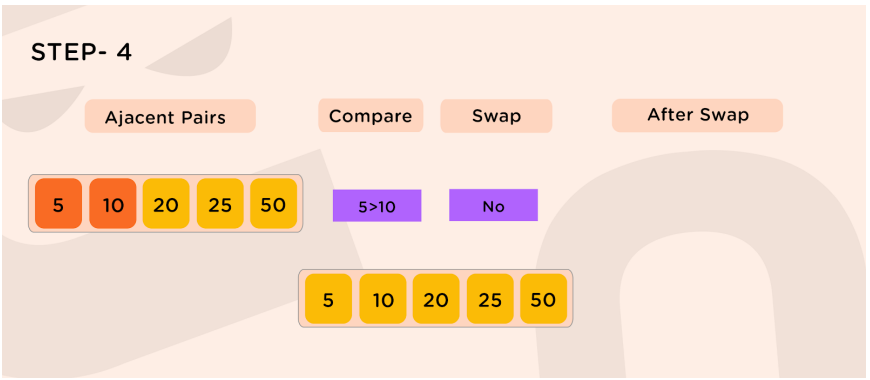
50	25	5	20	10

Following is the pictorial diagram for a better explanation of how it works:









Pseudocode:

```
/*
      array of size N from 0 to N-1 is considered
*/
function bubbleSort(arr, N)

          for idx = 0 to N-2
                    //   Last idx elements are already sorted
                    for jdx = 0 to N-idx-2
```

```
if arr[jdx] > arr[jdx+1]
        swap(arr[jdx],arr[jdx+1])
```

Time complexity: O(N^2), in the worst case.
For every element, we iterate over the entire array each time giving an overall time complexity of O(N^2).
Space complexity: O(1), as no extra space is required.