# **Using Iterative way**

Sort an array without using any inbuilt function by iterative way.

# C#

## Sorting an array of integers

1. Declaring and initializing the array
2. Creating a temporary variable to store the next value
3. Creating a loop which will iterate through the entire array until it reaches the second last position
4. Creating a loop inside that loop which will iterate from the second variable to the end of the array
5. So for loop 1, i=0 and j=1, therefore when we condition the loop, we are asking if the first element of the array

arr[i] = arr[0]

is smaller than the second element of the array

arr[j] = arr[1]

Therefore in our array it would be comparing

N0 N1 N2 N3 N4 N5

{1 9 6 7 5 9}

Arr[0] = 1

Arr[1] = 9

And since 1 is smaller than 9, we initiate the loop.

We need to swap the 1 and the 9 to make this into descending order

1. We store the value of arr[i] = arr[0] = 1

In a temporary variable called temp

Temp holds the value of 1 in this instance

1. We now store the value of arr[j] = arr[1] = 9 inside of where arr[i] = arr[0] had the 1

So now our array looks as follow

N0 N1 N2 N3 N4 N5

{9 9 6 7 5 9}

And the value of the 1 is held in the temp variable

1. Now we take arr[j] = arr[1] = 9 and we move the value of the temporary variable (which came from the first element of the array)

Now our array looks as follows

N0 N1 N2 N3 N4 N5

{9 1 6 7 5 9}

And we are closer to achieving descending order.

The loops will continue until eventually we have a loop which looks as follows

N0 N1 N2 N3 N4 N5

{9 9 7 6 5 1}

1. Condition for the first break = j < arr.Length[6]

As long as j is smaller than 6

2. Condition for second break = i < [arr.Length – 1]5

As long as I is smaller than 5

3. If condition: if it is smaller than the next

# Debugger

Loop 1 outer, loop 1 inner:

i=0, j=1, therefore we compare to see if position 0 is smaller than position 1 in value

N0 N1 N2 N3 N4 N5

{9 1 6 7 5 9}

Which it is not, and therefore the if statement will not execute and the array will remain the same.

## Loop 1 outer, loop 2 inner:

i=0, j=2, therefore we compare to see if position 0 is smaller than position 2 which is not true and the if statement will not execute, we know the next four loops will also not execute the if statement due to 9 being the biggest number. So after exiting with the loop remaining the same we move onto the second outer loop.

Loop 2 outer, loop 1 inner:

i=1, j=2, therefore we compare to see if position 1 is smaller than position 2 in value which is true, the if loop will execute and swap the values in position 1 and 2. Temp will be 1.

N0 N1 N2 N3 N4 N5

{9 6 1 7 5 9}

## Loop 2 outer, loop 2 inner:

i=1, j=3, in this case N1 is smaller than N3 and we swap them. Temp will be 6

N0 N1 N2 N3 N4 N5

{9 7 1 6 5 9}

## Loop 2 outer, loop 3 inner:

i=1, j=4, in this case no execution of if statement

N0 N1 N2 N3 N4 N5

{9 7 1 6 5 9}

## Loop 2 outer, loop 4 inner:

i=1, j=5, in this case N1 is smaller than N5 and they swap. Temp is 7.

N0 N1 N2 N3 N4 N5

{9 9 1 6 5 7}

## Loop 3 outer, loop 1 inner:

i=2, j=3, N2 is smaller than N3 so they swap. Temp is 1.

N0 N1 N2 N3 N4 N5

{9 9 6 1 5 7}

## Loop 3 outer, loop 2 inner:

i=2, j=4, no execution of if statement

N0 N1 N2 N3 N4 N5

{9 9 6 1 5 7}

## Loop 3 outer, loop 3 inner:

i=2, j=5, N2 is smaller than N5 so they swap. Temp is 6.

N0 N1 N2 N3 N4 N5

{9 9 7 1 5 6}

## Loop 4 outer, loop 1 inner:

i=3, j=4, N3 is smaller than N4 so they swap. Temp is 1.

N0 N1 N2 N3 N4 N5

{9 9 7 5 1 6}

## Loop 4 outer, Loop 2 inner:

i=3, j=5, N3 is smaller than N5 so they swap. Temp is 5.

N0 N1 N2 N3 N4 N5

{9 9 7 6 1 5}

## Loop 5 outer, Loop 1 inner:

i=4, j=5, N4 is smaller than N5 so they swap. Temp is again 1.

N0 N1 N2 N3 N4 N5

{9 9 7 6 5 1}

# JavaScript

## Sorting an array of integers

1. Create a function called sortArrays which takes an array as a parameter
2. Find the length of the array and store it in a variable called length
3. Using a for loop, we will loop until the second last element and increment j after every loop
4. If the item is smaller than the next item, we must swap them using a temp variable
5. And every time the if statement executes the value of j will be set to -1 to let the loop restart until every element is in order, because in order for the loop to end we need to increment the j
6. We will return the array to the function
7. Use the function

N0 N1 N2 N3 N4 N5

{9 1 6 7 5 9}

1. **As long as j is not bigger than 5, the loop will continue**
2. **If the array element is smaller than the next, they will swap**

# Loop1

J=0, j+1=1, so we compare element N0 and N1, and can see N0 is bigger therefore the if wont execute and it increases J by 1

# Loop2

J=1, j+1=2, so we compare element N1 and N2, since N1 is smaller, they swap, Temp is 1. And J becomes -1

N0 N1 N2 N3 N4 N5

{9 6 1 7 5 9}

Since J is now -1 and the loop is over, j gets increased to 0. And a new loop starts

# Loop3

J=0, j+1=1, If not executed

N0 N1 N2 N3 N4 N5

{9 6 1 7 5 9}

# Loop4

J=1,j+1=2, if not executed

N0 N1 N2 N3 N4 N5

{9 6 1 7 5 9}

# Loop5

J=2,j+1=3, N2 is smaller than N3 so it swaps. Temp is 1. And J becomes -1, gets 1 added to become 0 and a new loop starts

N0 N1 N2 N3 N4 N5

{9 6 7 1 5 9}

# Loop6

J=0,j+1=1, N/E

N0 N1 N2 N3 N4 N5

{9 6 7 1 5 9}

# Loop7

J=1,j+1=2, N1 is smaller than N2 so it swaps. Temp is 6. J becomes -1, gets 1 added to become 0 and a new loop starts

N0 N1 N2 N3 N4 N5

{9 7 6 1 5 9}

# Loop8

J=0,j+1=1, N/E

N0 N1 N2 N3 N4 N5

{9 7 6 1 5 9}

# Loop9

J=1,j+1=2, N/E

N0 N1 N2 N3 N4 N5

{9 7 6 1 5 9}

# Loop10

J=2,j+1=3, N/E

N0 N1 N2 N3 N4 N5

{9 7 6 1 5 9}

# Loop11

J=3,j+1=4, N3 is smaller than N4 so it swaps. Temp is 1. J becomes -1, gets 1 added to become 0 and a new loop starts

N0 N1 N2 N3 N4 N5

{9 7 6 1 5 9}

# Loop12

J=0,j+1=1, N/E

N0 N1 N2 N3 N4 N5

{9 7 6 5 1 9}

# Loop13

J=1,j+1=2, N/E

N0 N1 N2 N3 N4 N5

{9 7 6 5 1 9}

# Loop14

J=2,j+1=3, N/E

N0 N1 N2 N3 N4 N5

{9 7 6 5 1 9}

# Loop15

J=3,j+1=4, N/E

N0 N1 N2 N3 N4 N5

{9 7 6 5 1 9}

# Loop16

J=4,j+1=5, N4 is smaller than N5 so it swaps. Temp is 1. J becomes -1, gets 1 added to become 0 and a new loop starts

N0 N1 N2 N3 N4 N5

{9 7 6 5 1 9}

# Loop17

J=0,j+1=1, N/E

N0 N1 N2 N3 N4 N5

{9 7 6 5 9 1}

# Loop18

J=1,j+1=2, N/E

N0 N1 N2 N3 N4 N5

{9 7 6 5 9 1}

# Loop19

J=2,j+1=3, N/E

N0 N1 N2 N3 N4 N5

{9 7 6 5 9 1}

# Loop20

J=3,j+1=4, N3 is smaller than N4 so it swaps. Temp is 1. J becomes -1, gets 1 added to become 0 and a new loop starts

N0 N1 N2 N3 N4 N5

{9 7 6 9 5 1}

# Loop21

J=0,j+1=1, N/E

N0 N1 N2 N3 N4 N5

{9 7 6 9 5 1}

# Loop22

J=1,j+1=2, N/E

N0 N1 N2 N3 N4 N5

{9 7 6 9 5 1}

# Loop23

J=2,j+1=3, N2 is smaller than N3 so it swaps. Temp is 6. J becomes -1, gets 1 added to become 0 and a new loop starts

N0 N1 N2 N3 N4 N5

{9 7 6 9 5 1}

# Loop24

J=0,j+1=1, N/E

N0 N1 N2 N3 N4 N5

{9 7 9 6 5 1}

# Loop25

J=1,j+1=2, N1 is smaller than N2 so it swaps. Temp is 7. J becomes -1, gets 1 added to become 0 and a new loop starts

# Loop26

J=0,j+1=1, N/E

N0 N1 N2 N3 N4 N5

{9 9 7 6 5 1}

# Loop27

J=1,j+1=2, N/E

N0 N1 N2 N3 N4 N5

{9 9 7 6 5 1}

# Loop28

J=2,j+1=3, N/E

N0 N1 N2 N3 N4 N5

{9 9 7 6 5 1}

# Loop29

J=3,j+1=4, N/E

N0 N1 N2 N3 N4 N5

{9 9 7 6 5 1}

# Loop30

J=4,j+1=5, N/E

N0 N1 N2 N3 N4 N5

{9 9 7 6 5 1}

# Loop31

J=5,j+1=6, now that j is not smaller than 5, It is equal to 5, the for loop will exit and return the loop to the function for outputting

N0 N1 N2 N3 N4 N5

{9 9 7 6 5 1}

N0 N1 N2 N3 N4 N5

{1 9 6 7 5 9}

## Sorting an array of strings