

Name : Dhruva Saha

Roll No: B.Sc (Sem-~~IV~~ IV) - 04

Program : Write a program in C to sort a given list of n numbers in ~~deceen~~ descending order using the bubble sort method, recursively

Date : 30/06/2022

## Implement Bubble Sort using Recursion

### ~~Algorithm~~ Algorithm

Let assume that we have an unsorted array \*arr containing  $n$  number of elements

Step 1. IF size of the array is 1, then the array is already sorted and return

Step 2. ELSE perform one loop of iterative bubble sort on the sub-array. It will place the last element at its correct place.

Step 3. Using recursion call step 1 & step 2 will perform repeatedly unless the size of the ~~arr~~ sub-array became one. With every recursion call it will reduce size ~~of~~ of the sub-array by one.

Day Run

Here  $n = 4$  (as per code)

The array =  $(20, 10, 30, 5)$

At 1st iteration

$(\underline{20}, 10, 30, 5) \rightarrow (20, 10, 30, 5)$

No change occurred as  $20 > 10$

$(20, \underline{10}, 30, 5) \rightarrow (20, 30, 10, 5)$

10 & 30 swapped as  $10 < 30$

$(20, 30, \underline{10}, 5) \rightarrow (20, 30, 10, 5)$

No change occurred as  $10 > 5$

At 2nd iteration

$(20, 30, 10, 5) \rightarrow (30, 20, 10, 5)$

20 & 30 swapped as  $20 < 30$

$(30, \underline{20}, 10, 5) \rightarrow (30, 20, 10, 5)$

No change occurred as  $20 > 10$

At 3rd iteration

$(30, 20, 10, 5) \rightarrow (30, 20, 10, 5)$

No change occurred as  $30 > 10$

## Time Complexity

### Average Case

On average case,  $n-i$  comparison needed for  $i$ th pass of bubble sort. For  $n$  number of pass

$$(n-1) + (n-2) + (n-3) + \dots + 1 = \frac{n(n-1)}{2}$$

∴ On average case time complexity is order of  $O(n^2)$

### Best Case

~~The array is~~ On best case ~~the array is already sorted~~, the array is already sorted

∴ On best case time complexity is order of  $O(n)$

### Worst Case

Worst case occurs when the array is reversely sorted, so max number of comparisons and swapping need to be performed

∴ On worst case time complexity is order of  $O(n^2)$