

# Bubble Sort

Vivek K. S., Deepak G.

Information Systems Decision Sciences (ISDS)  
MUMA College of Business  
University of South Florida  
Tampa, Florida

2017

# Bubble Sort

- Bubble sort is based on comparison of adjacent items in a list that swaps the items to place them in the proper order.
- It is done through multiple passes, because an item is swapped only to the next position in the list based on the result of the comparison operation (and not all the way to the end).
- Each pass, achieves placing the next largest item in the list at its proper position.
- In the first pass, in a list of  $n$  items, there are  $n-1$  pairs of items to process.
- In the second pass, since the largest number has already been placed in its proper position, we are left with  $n-1$  items and  $n-2$  pairs of items. This is how it progresses. <sup>1</sup>

---

<sup>1</sup>Animation available at

# Python Implementation

```
def bubble_sort(a_list):  
    for passes in range(len(a_list) - 1, 0, -1):  
        for i in range(passes):  
            if a_list[i] > a_list[i + 1]:  
                a_list[i], a_list[i + 1] =  
                    a_list[i + 1], a_list[i]  
    return(a_list)  
a = [1,2,3,4,6,345,25,6,25,5,6,72,61,6,262]  
bubble_sort(a)  
[1, 2, 3, 4, 5, 6, 6, 6, 6, 25, 25, 61, 72, 262, 345]
```

Notice that we used the simple two-variable swap that Python offers to swap the list items here.

# Explanation

- The number of passes is decided by the for loop that runs on a negative-stepping range.
- The first pass is based on the full length of the list.
- After the first pass, the number of items in the list on which the sorting has to operate keeps decreasing.
- In every pass, the adjacent items are compared against each other and are swapped based on the result.
- Items that have the same value are not swapped and they appear together in the end result.

# Analysis of Bubble Sort

- Regardless of the arrangement of the items in the list initially,  $n-1$  passes will be needed to sort the list of size  $n$ .
- The total number of comparisons is the sum of first  $n-1$  integers.
- Using the formula for the sum of first  $n$  integers, we find that the sum of  $n-1$  integers equals  $1/2n^2 - 1/2(n-n)$  which is  $n^2/2 + n/2$ .
- This is  $O(n^2)$  comparisons.
- In the best case, the list is already sorted. But in the worst case, every comparison will result in a swap.
- Bubble sort is the most inefficient way to sort a collection of items.

# Summary

- Bubble sort is one of the most fundamental sorting algorithms in Computer Science.
- It sorts items through repeated passes, sorting one element at a time.
- It offers good performance for smaller problems but as the number of items go up, the performance goes down.
- It is evident from the fact that it's Big-O Notation is  $O(n^2)$ .
- But it is the most basic and easy to use algorithm as it is very straightforward in its working.