DSME

Bubble Sort

Data Structures Made Easy

DUBLIN CITY UNIVERSITY

Contents

1. *Definition*

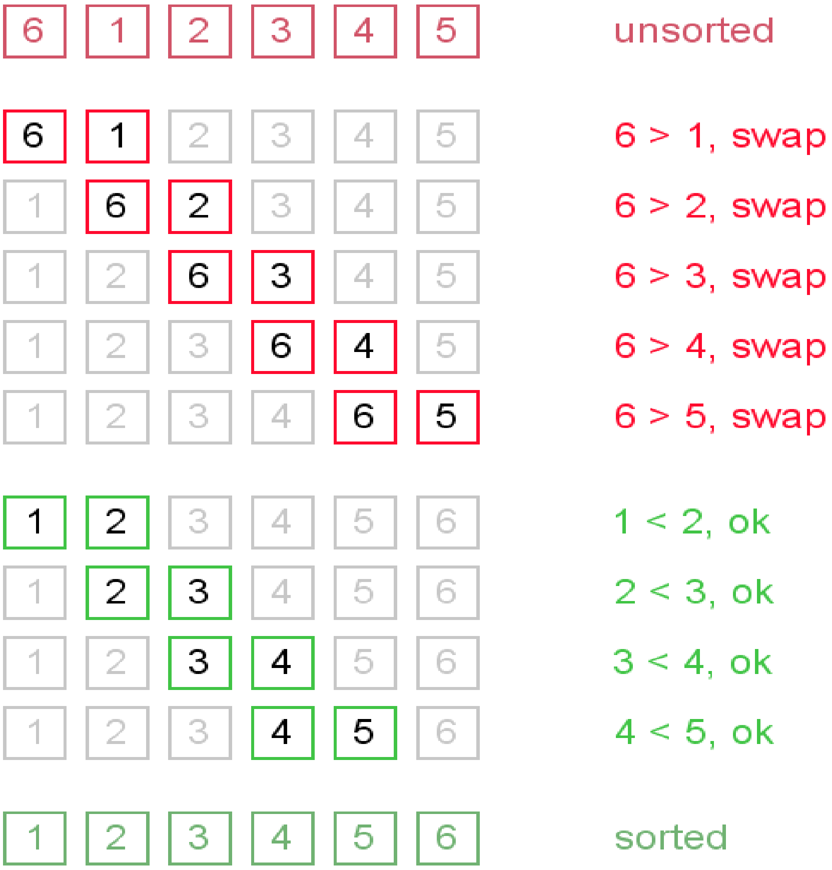
Bubble sort is a comparison sorting algorithm that computes by repeatedly analysing lists that need to be sorted. Each pair of adjacent elements are compared and swapped if they are in the incorrect order. This procedure is repeated until no more swaps are required, thus indicating that the list is correctly sorted.

2. *Implementation*

Bubble sort works as follows:

1. Take the first two elements of array and place the larger element at the higher index.
2. Repeat process is repeated until the largest element is not reached at the final index.
3. Start again at first index of the array up to n-1 index.
4. Process is repeated until all elements are correctly sorted.

3. *Example*



6 1 2 3 4 5 unsorted

6 1 2 3 4 5 6 > 1, swap

1 6 2 3 4 5 6 > 2, swap

1 2 6 3 4 5 6 > 3, swap

1 2 3 6 4 5 6 > 4, swap

1 2 3 4 6 5 6 > 5, swap

1 2 3 4 5 6 1 < 2, ok

1 2 3 4 5 6 2 < 3, ok

1 2 3 4 5 6 3 < 4, ok

1 2 3 4 5 6 4 < 5, ok

1 2 3 4 5 6 sorted

4. *Functions*

The implementation of bubble sort is based on the following functions

- **bubble()**
  This function takes the unsorted array and traverses it from 0 to the length-1 index of the array. The selected element is compared to the next element and the elements are swapped if the next element is of a lesser value. The largest value is eventually placed at the largest index. This process repeats until all the elements are correctly sorted.

5. *Pseudocode*

```
function bubble ( array arr, size ){

        for index_1 ( 0 to size of arr )

                for index_2 ( 1 to size of arr – index_1 )

                        if index_2 – 1 of arr > index_2 of arr
                                swap ( arr, index_2 – 1, index_2 )
}
```

6. *Complexity*

The optimal time complexity of bubble sort is O(N). This performance can only be achieved over a currently sorted list.

Bubble sort has a average and worst-case time complexity of O(N ^ 2). N is based on the amount of sorted items. The positioning of the elements determines the performance of the algorithm. A large element at the beginning of the list does not affect the performance of the algorithm. However, small elements at the end of the list take a considerable amount of swapping time, thus affecting the time performance.

7. ***Advantages of Bubble Sort***

Bubble sort is a simplistic algorithm to implement. There is minimal overhead due to data being sorted in-place. When data is correctly sorted, the data is in memory and ready to be processed efficiently.

8. ***Disadvantages of Bubble Sort***

Bubble sort's computational time complexity is the source of its disadvantages. The average time of completion increases vastly as the number of unsorted elements within the list increases. The sorting process requires multiple passes over the data, which imposes an inefficient completion time.

9. ***References***

Janssen C. 2013. *Bubble Sort* [Online]. Available from:
http://www.techopedia.com/definition/3757/bubble-sort

Leander R. 2011. *Advantages & Disadvantages of Bubble Sort* [Online]. Available from:
http://www.ehow.com/info_8201350_advantages-disadvantages-bubble-sort.html

Rose India 2007. *Bubble Sorting in Java* [Online]. Available from:
http://www.roseindia.net/java/beginners/arrayexamples/bubblesort.shtml