Bubble Sort Algorithm

Sultan Altukhaim
Florida Institute of Technology
Department of Computer Sciences
Melbourne, Florida, USA
saltukha@fit.edu

Abstract – A Sorting algorithm is an algorithm that arranges a group of elements of a list in a certain order. Sorting algorithms are taught in a few fields such as Computer Science and Mathematics. They differ in their complexity, functionality, performance, application, and resources usage. In this short paper, we briefly shed the light on an old and common sorting algorithm, Bubble Sort.

1 Definition

Bubble sort is a simple technique of sorting an array’s elements. The algorithm starts with the first elements of the array (data set). When set in an ascending order, it compares the first two items in the array, and if the first item is greater than the second one, it swaps them. Then, it goes on performing this process for each pair of adjacent items until it reaches the end of the data set. It then begins again with the first two items, iterating the same comparison process until no swaps have happened on the last pass (in other words, all items are in the correct order). The algorithm’s name, bubble sort, comes from a natural water phenomenon where larger items sink to the end of the list whereas smaller values “bubble” up to the top of the data set.

2 Performance

Although this algorithm is simple, it is greatly inefficient and is seldom implemented in real application except in theory and education. For example, if we have 100 array’s items then the number of passes is going to be 10000. The locations of the data items in bubble sort play a substantial role in determining its performance. Large elements at the beginning of the list do not cause an issue, as they can be quickly swapped. However, small elements towards the end of the list can be brought to the beginning exceptionally slowly. There exist some other sorting algorithms that have better efficiency than bubble sort does, such as Insertion Sort, Merge Sort, and Selection Sort, which are out of this paper’s scope.

3 Example

Given an array 34265 a bubble sort algorithm would generate the following sequence of semi-sorted arrays: 32465, 32456, 23456. First the 2 and 4 would be compared and swapped, then the 5 and 6. On the next pass, the 2 and 3 would switch, and the data set would be in order.

4 Advantages and drawbacks

The decision of using an appropriate sorting algorithm should depend on some metrics and criteria that concern data set size, resources, algorithm complexity, and timeliness, because each sorting algorithm has its own strengths and weaknesses. Speaking of advantages, bubble sort enjoys many: simple to program, easy to understand, quick to program, and low risks of having bugs. On the other hand, its only drawback, inefficiency, negatively affects its usability. Therefore, it is still debatable among programmers whether or not to continue using bubble sort in practice.