Bubble Sort

Jerry Zhu

What is Bubble Sort?

- Referred to as “sinking sort”, it is the simplest sorting algorithm.
- Can be treated as an introduction into sorting algorithms, and a stepping stone to learning more complex sorting methods.
- It is called bubble sort, because, over time, each element will “rise” to their corresponding positions in the array, like bubbles.
- Compared to other algorithms, bubble sort is extremely inefficient.
- It has worst-case(when items are in reverse order) and average complexity of O (n2).
- It does have O(n) for its best case, but that is only if the array is sorted already.

How Does It Work?

- Given an array of elements, it compares, and if their order is incorrect, swaps the first two values. Then, the next two adjacent values are compared, and so on, until the end of the array is reached.
- The sort will start again from the first element, and repeat the sort until no further swaps are necessary, in which the array will then be sorted.

How Does It Work? Cont.

- To keep track of whether a swap has been made during a sorting “lap”, bubble sort also uses a boolean variable.
- This boolean variable is usually set to either true or false before each “lap”, and will be set to the opposite value at the same time a swap is made.
- That way, at the end of each run-through, if the boolean value is still the same as it was before the “lap”, the program will know that the array is now sorted, and no more “laps” are necessary.
- However, if the boolean value has changed, the program will have to make another “lap”, to check whether the elements are now sorted.

Cautions

- While it is one of the easiest sorting algorithms to implement, this comparison sort’s inefficiency makes it too slow and too impractical for most situations.
- Consider this real-world problem: A census is conducted by the Ontario provincial government and you, the government’s only computer scientist, is asked to sort all of the people by age. Using bubble sort is undesirable here because the sample is massive and random. It would take a long time for the computer to sort through the millions of people. A more efficient sort for processing such large data would be merge sort or quick sort.

4 Tips for Success

1. Remember your ‘temp’ variable.
2. Remember that you need TWO for loops.
3. Remember to adjust the indexes of the second for loop so that the sort doesn’t throw an ArrayIndexOutOfBounds Exception.
4. Remember that elements that aren’t set explicitly are given a default value of 0 so you won’t have to worry about a NullPointerException.

Personal Reflection

- Through the making of the bubble sort presentation, I gained a robust understanding of a fundamental sorting algorithm that serves as a benchmark on which I can learn higher-level algorithms.
- Other than learning the concepts, I now have a greater appreciation for the power of algorithms and the things they can do. When applied properly, it has endless possibilities.