Diksha Sharma

Task1

**Bubble sort**

Bubble sort is a sorting algorithm that works by repeatedly stepping through lists that need to be sorted. It is also known as sinking sort. It cycles through a list, analyzing pairs of elements from left to right, or beginning to end. If the leftmost element in the pair is less than the rightmost element, the pair will remain in that order. If the rightmost element is less than the leftmost element, then the two elements will be switched. This cycle repeats from beginning to end until pass in which no switch occurs. This algorithms best case performance is O (n) and the average and worst case performance is O (n2), so it is rarely used to sort large, unordered data sets. Some of the properties of bubble sort algorithm are that it is stable, has O (1) extra space, O (n2) comparisons and swaps and its adaptive: O (n) when nearly sorted. Some of the advantages of bubble sort algorithm are that it is simple, easy to understand, little memory overhead. The disadvantages of this algorithm are that it is time consuming, and it is slow compared to other sorting methods. Bubble sort algorithm is not efficient in case of reverse ordered collection.

 **Bubble sort**

**Psedudo-code**

func bubblesort( var a as array )

for i from 1 to N

for j from 0 to N - 1

if a[j] > a[j + 1]

swap( a[j], a[j + 1] )

end func

**How many comparisons will be needed to sort the array?**

The formula used is n-1.  Where n = the number of elements in the array.  So, if you have 55 elements, you will have to compare 55-1 times

Sorting: 10, 1, 3, 9, 8

1 10 3 9 8

1 3 10 9 8

1 3 9 10 8

1 398 10

1 3 8 9 10

Total numbers of swaps are 5 and total numbers of comparisons are 4.

Sorting: 10, 9, 8, 3, 1

9 10 8 3 1

9 8 10 3 1

9 8 3 10 1

9 8 3 1 10

8 9 3 1 10

8 3 9 1 10

8 3 1 9 10

3 8 1 9 10

3 1 8 9 10

1 3 8 9 10

Total numbers of swaps are 10 and total numbers of comparisons are 4. If the sort always pivots on the beginning element (index 0) the sort tree will bigger (more level) than if it always pivots on the next element after the beginning one(index 1).

References

[Online] 2007. <http://www.sorting-algorithms.com/bubble-sort> (accessed: 20 February 2014).

[Online] 2008. <http://pkirs.utep.edu/cis3355/Tutorials/chapter9/tutorial9A/bubblesort.htm> (accessed: 19 February 2014)

[Online] 2009. <http://www.youtube.com/watch?v=P00xJgWzz2c> (accessed: 20 February 2014).