ADS Homework 6

Sanjay Timilsena

March 2020

1 Problem 6.1

1.1 a)

```
void bubble_sort(array[], size){
    //outer loop that goes through each element
    for(j=0 to size-1){
    //flag to monitor swapping
    flag=0;
    //inner loop that goes through non-sorted elements
        for(i=0 to size-i-1){
        //when prev element is larger than next, swap them
            if(array[i] > array[i+1]){
                swap array[i] & array[i+1];
                flag=1;
            }
        }
        //when no swapping occurs, array is sorted
        if(!flag){
            break;
        }
    }
}
```

1.2 b)

Worst-Case: When the number are arranged in opposite order, it takes the maximum number of swaps and loops to completely sort the array. The number of swaps decreases in each loop from (n-1) to 1. So, the total time taken for swaps can be calculated as:

$$T(n) = \sum_{1}^{n-1} n$$

Hence we get the time complexity,

$$T(n) = O(n^2)$$

Average-Case: The average case may contain random number of swaps and the number of swaps involved provides a constant term that divides the time complexity of worst-case. For example, if the number of swaps is half of that of worst-case, then the time complexity becomes $\theta(\frac{n^2}{2})$ which corresponds to $\theta(n^2)$. So, the time complexity for average case is given by:

$$T(n) = \theta(n^2)$$

Best-Case: The minimum number of swaps takes place when the array is already sorted. In this case, the inner loop runs from 0 to n-2, while the outer loop runs only once. So, the time complexity for best-case is given by:

$$T(n) = \Omega(n)$$

1.3 c)

The sorting algorithms Insertion Sort, Merge Sort and Bubble Sort are stable as two records R and S with the same key and with R appearing before S in the original list, R will appear before S in the sorted list while sorting from this algorithms. However, Heap Sort is not a stable algorithm because it is not arranged linearly and a record that appears after another record with same key may reach to the top/left branch of tree which changes the original order of this records.

1.4 d)

The sorting algorithms Insertion sort and Bubble sort have different time complexities for best and worst cases so they are adaptive. The sorting algorithms Merge sort and Heap sort have same time complexities for best and worst cases so they are non-adaptive.

2 Problem 6.2

2.1 a)

Please see the attached file heap_sort.cpp.

2.2 b)

Please see the attached file heap_sort_variant.c. Source: 'geeksforgeeks'

2.3 c)

Heap sort variant is faster for smaller size of input while Heap sort is faster for larger size of input.