BLG 335E – Analysis of Algorithm I, Fall 2017
Project 2 Report

Assignment Date: 19 Oct 2017 Thursday

Due Date: 09 Nov 2017 Thursday – 23:59

Kadir Emre Oto

150140032

Code Analysis

In this assignment, we are expected to implement quick sort and measure the elapsed time. All codes are written in **kod.cpp** file.

Compilation Command: g++ kod.cpp -o kod -O2 -std=c++11
Running Command: ./kod 10000

a. (10 Points) Give the asymptotic upper bound on the running time for Quicksort. If there is a recurrence for the algorithm, give and solve it.

There are two recurrence equation for Quicksort algorithm:

Best and average case: (if the input distributed randomly)

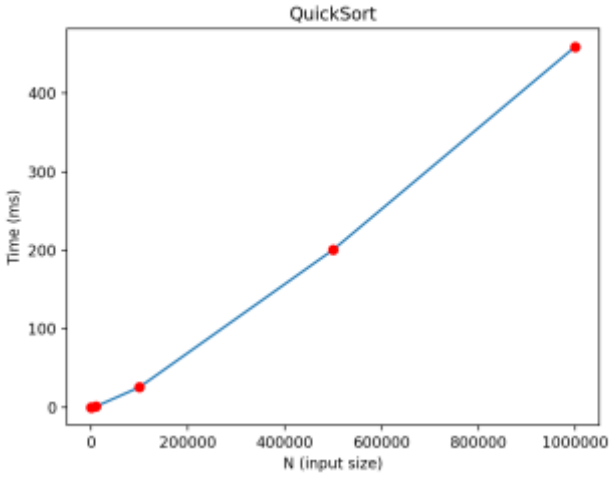
$$T(n) = 2 * T(n/2) + n$$
$$n = \theta\left(n^{\log_2 2}\right) \rightarrow T(n) = \theta\left(n^{\log_2 2} * \log_2 n\right) = \theta(n * \log_2 n) \text{ (master thm. case 2)}$$

Worst case: (if the input is sorted or reversed sorted)

$$T(n) = T(n-1) + n$$
$$T(n-1) = T(n-2) + n - 1$$
$$T(n-2) = T(n-3) + n - 2$$
$$\dots$$
$$T(1) = 1$$
$$T(n) = \frac{n * (n+1)}{2} = O(n^2)$$

b. (10 Points) Run your program for **N** as {10, 100, 10000, 100000, 500000, 1000000}, and calculate the average time of execution (for example run 10 times for **N** = 10 and take the average, 10 times for **N** = 100 and take the average, and so on…) for each of **N**. After calculating average execution times, you are required to prepare a two-lined plot in order to visualize the run time complexity of Quicksort for different values of **N**. Comment on the results by considering the asymptotic bound that you have found in ***(a)***.

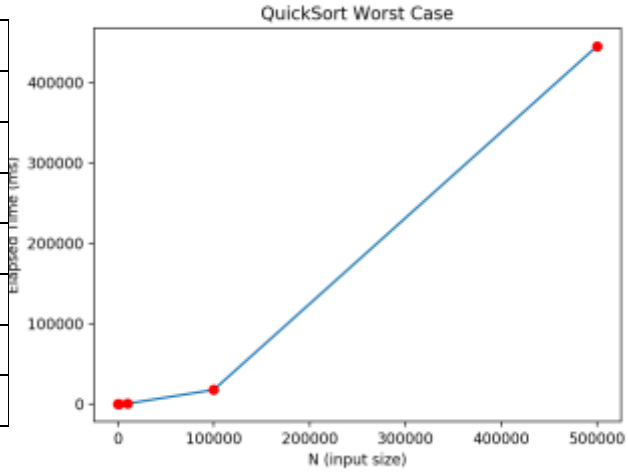
N (input size)	Elapsed Time
10	0 ms
100	0 ms
1000	0 ms
10000	1.05 ms
100000	25.60 ms
500000	200.50 ms
1000000	459.00 ms



c. (20 Points) What is the worst case for Quicksort? Construct a data set in order to simulate this case, and execute your program for different values of N as {10, 100, 1000, 10000, 100000, 500000, 1000000}. Report the average execution times, and introduce a plot like you have done in part (b). Introduce a solution in order to overcome the worst case of Quicksort (You do not need to implement that solution)

Two cases are the worst case for Quicksort: input is sorted, and input is reversed sorted. So we can use the sorted version of given input for assignment, which is output of the program we are expected to code. But because the worst case complexity of Quicksort is $O(n^2)$, we are not able to run the program for N = {500000, 1000000}. To handle the worst cases, we can choose the pivot randomly in partition part of quicksort, it reduces the complexity to $\theta(n * \log_2 n)$ again.

N (input size)	Elapsed Time
10	0 ms
100	0 ms
1000	5.3 ms
10000	338.58 ms
100000	17523.60 ms
500000	445099.43 ms
1000000	too large



The following code handles the worst case scenario for Quicksort by selecting pivot randomly:

```
void Population::QuickSort(Residence** array, int length){
    if (length < 2)
        return;

    int sep=1;
    swap(array[0], array[rand() % length]);  // for sellection random pivot

    for (int i=1; i < length; i++)
        if (*(array[i]) < *(array[0]))
            swap(array[sep++], array[i]);

    swap(array[0], array[sep-1]);
    QuickSort(array, sep-1);
    QuickSort(array+sep, length-sep);
}
```

d. (10 Points) Is Quicksort stable? Explain your answer by illustrating your answer with a small fraction from data set. You may modify the values if necessary.

Quicksort is **not** a stable sorting algorithm by default, maybe the initial position of elements can be stored and considered in comparing step. Following example explains why quicksort is not stable:

Pivot i, j
(4,1,2,male,1,1), (5,1,2,female,1,1), (5,1,2,male,1,1), (1,1,2,male,1,1)

Pivot i, j
(4,1,2,male,1,1), (5,1,2,female,1,1), (5,1,2,male,1,1), (1,1,2,male,1,1)

Pivot i, j
(4,1,2,male,1,1), (5,1,2,female,1,1), (5,1,2,male,1,1), (1,1,2,male,1,1)

Pivot i j
(4,1,2,male,1,1), (1,1,2,male,1,1), (5,1,2,male,1,1), (5,1,2,female,1,1)

At last step, (5,1,2,female,1,1) is swapped with (1,1,2,male,1,1) and the order between (5,1,2,female,1,1), (5,1,2,male,1,1) is clearly ruined. It means the algorithm is not stable.