ADS Homework 3

Sanjay Timilsena

February 2020

1 Problem 3.1

1.1 a)

Here,

$$f(n) = 9n, \qquad g(n) = 3n^2$$

Using definition from limits,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{9n}{3n^2} = 0$$

It implies,
$f \in O(g)$, $f \in o(g)$, $g \in \Omega(f)$, $g \in \omega(f)$

1.2 b)

Here,

$$f(n) = 9n^{0.8} + 2n^{0.3} + 14\log n, \qquad g(n) = \sqrt{n}$$

Using definition from limits,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{9n^{0.8} + 2n^{0.3} + 14\log n}{\sqrt{n}} = \infty$$

It implies,
$f \in \Omega(g)$, $f \in \omega(g)$, $g \in O(f)$, $g \in o(f)$

1.3 c)

Here,

$$f(n) = \frac{n^2}{\log n}, \qquad g(n) = n\log n$$

Using definition from limits,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{\frac{n^2}{\log n}}{n\log n} = \infty$$

It implies,
$f \in \Omega(g)$, $f \in \omega(g)$, $g \in O(f)$, $g \in o(f)$

1.4 d)

Here,

$$f(n) = (\log n)^3, \qquad g(n) = 9 \log n$$

Using definition from limits,

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \frac{(\log n)^3}{9 \log n} = \infty$$

It implies,
$f \in \Omega(g)$, $f \in \omega(g)$, $g \in O(f)$, $g \in o(f)$

2 Problem 3.2

2.1 a)

```

void Selection_sort(int Arr[], int n){
    int holder, key, pos;
    for(int j=0; j<(n-1); j++){
        holder = Arr[j];
        key = Arr[j+1];
        for(int i=j+2; i<n; i++){
            if(Arr[i] < key){
                key = Arr[i];
                pos = i;
            }
        }
        Arr[j] = key;
        Arr[pos] = holder;
    }
}
```

2.2 b)

For the above Selection-sort function, the outer loop shows **loop invariant**. It can be described by the fact that before and after each inner loop the array upto current position is always sorted.

Maintenance

The inner loop searches for the smallest value in the array A[j+2...n]. The current value determined by the outer loop, A[j] is replaced by the smallest value. Then the container containing the smallest value previously is replaced by the last value in A[j]. In this way A[0...j] is always sorted.

Termination

When j moves from 0 to n-1 it keeps on sorting A[0...j] and finally the whole A[0..(n-1)] is sorted when the loop terminates.

2.3 c)

Here the solution program for generation of random sequence for Cases A and B with average case is attached as Selection-sort.cpp. The program contains the following cases as:

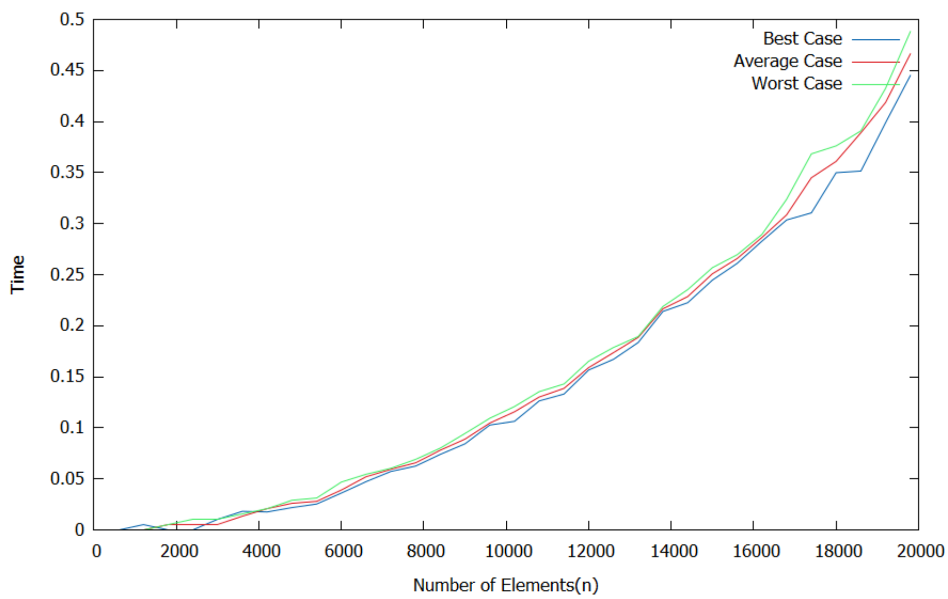
Case A: The case with most swaps. In the program, it is defined by the function **int* worstCase(int n);** in which the elements of the array are ordered as highest, lowest, 2nd highest, 2nd lowest.. and so on. This generates a sequence which requires most swaps in order to be sorted.

Case B: The case with least swaps. In the program, it is defined by the function **int* bestCase(int n);** in which the elements of a randomly generated array are already sorted.

Average Case: The case with random number of swaps. In the program, it is defined by the function **int* averageCase(int n);** which creates an array with n number of integers in random order.

2.4 d)

The curve plotted using Gnuplot for the file data.txt exported by the program is given below. The file contains number of data, time taken for best, average and worst case respectively which is shown in the plot.



2.5 e)

From the plot, we can conclude that the time complexity for all cases is $O(n^2)$