Analysis of Algorithms

Advanced Algorithms – Assign#1 **Due date: Sept 7th**

1) Order the following functions by growth rate : N, $N^{1/2}$, $N^{1.5}$, N^2, NlogN, $N(\log N)^2$, $N\log N^2$, 2/N, 2N, $2^{N/2}$, 37, N^3, and $N^2\log N$. Also, Indicate which functions grow at the same rate.

$2/N < 37 < N^{1/2} < N = 2N < N\log N < N(\log N)^2 = N\log N^2 < N^{1.5} < N^2 < N^2\log N < N^3 < 2^{N/2}$

2) Describe the order of magnitude of each of the following code sections, using Big-O notation :

a)

```
static int Square_Root(int num) {
        int i = num;
        while(i * i >= num) {
                i = i - 1;
        }
        return (i + 1);
}
```

O(1) O(1) $\}$ $O(\sqrt{n})$ O(1)

b)

```
count = 0;
for(i = 1; i <= N; i++)
        count++;
for(j = 1; j <= N; j++)
        count++;
```

O(1) $\}$ O(n) $\}$ O(n) $= 2O(n) + O(1) = O(n)$

c)

```
value = N;
count = 0;
while (value > 1){
        value = value/2;
        count++;
}
```

O(1) O(1) $\}$ $O(\log n) = 2O(1) + O(\log n) = O(\log n)$

3) Consider four programs—A, B, C, and D—that have the following performances:
A - O(log *n*)
B - O(*n*)
C - O(n^2)
D - O(2^n)

in next page

If each program requires 10 seconds to solve a problem of size 1000, estimate the time required by each program for a problem of size 2000.

4) Calculating prefix average of a set of values is an important problem, especially in financial calculations. Use the following link to understand background on prefix averages problem - http://cs-fundamentals.com/tech-interview/dsa/prefix-averages-algorithm-java-program.php
Consider the algorithms (methods) – prefixAverage1 and prefixAverage2 for calculating prefix averages. The source code is given here. Implement both the algorithms in your choice of programming language and perform an experiment analysis of their running times under different

input sizes. Visualize the running times on a chart, where x-axis represents different input sizes and y-axis represents running times of the algorithms.

Submission:
Please upload your solution files to d2l.

3-) A-) $O(\log n)$

$10 \text{ sec} = C \frac{\text{sec}}{\text{operation}} \cdot \log(1000) \text{ operations}$

$\frac{10 \text{ sec}}{\log(1000) \text{ operations}} = C \frac{\text{sec}}{\text{operation}}$

$1.00343 \text{ sec/operation}$

$\Rightarrow X \text{ sec} = 1.00343 \cdot \log(2000)$

$X = 11.0034 \text{ seconds}$ to run a problem of size 2000

B-) $O(n)$ $10 \text{ sec} = C \frac{\text{sec}}{\text{operation}} \cdot 1000 \text{ operations}$

$C = \frac{10}{1000}$

$X \text{ sec} = \frac{10}{1000} \cdot 2000$

$X = 20 \text{ sec}$ to run a problem of size 2000

C-) $O(n^2)$

$10 \text{ sec} = C \frac{\text{sec}}{\text{operation}} \cdot (1000^2) \text{ operations}$

$C = \frac{10 \text{ sec}}{1000^2} \Rightarrow X \text{ sec} = \frac{10}{1000^2} \cdot 2000^2$

$X = 40 \text{ sec}$ to run a problem of size 2000

D-) $O(2^n)$

$10\ sec = C\ \frac{sec}{operation}\ (2^{1000})$

$C = \frac{10}{2^{1000}} \Rightarrow x\ sec = \frac{10}{2^{1000}} \cdot 2^{2000}$

$x = 10 \cdot 2^{1000}$ sec to run a problem of size 2000
$\approx 1 \cdot 10^{302}$

4-) code in separate file.