COSC 2100/2010 (Fall 2017)

Assignment II

Due by Tuesday, September 19, 2017 6:00 PM CST

**Submission**: Write your answers in this document. Do not delete the questions. Name the files as **FirstName\_LastName\_Project2.docx** then upload to D2L.

1. Sort the following functions by growth rate (i.e., order of complexity): N, N1/2, N1.5, N2, NlogN, N(logN)2, NlogN2, 2/N, 2N, 2N/2, 37, N3, and N2logN. Also, indicate which functions grow at the same rate.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Best** |  |  | **(same)** | **(same)** | **(same)** | **(same)** | **Worst** |
| **2/N** | **37** | **N1/2** | **N** | **NlogN2** | **N2** | **N3** | **2N/2** |
|  |  |  | **2N** | **NlogN** | **N2logN** | **N1.5** |  |
|  |  |  | **NlogN2** |  |  |  |  |

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

**a)**   **O (N)**

**static** **int** Square\_Root(**int** num) {

**int** i = num;

**while**(i \* i >= num) {

                     i = i - 1;

              }

**return** (i + 1);

       }

**b)**  **O (N)**

 From Text book 48 (e)

       count = 0;

**for**(i = 1; i <= N; i++)

              count++;

**for**(j = 1; j <= N; j++)

              count++;

**c)**   **O (log N)**

From Textbook 48 (c)

       value = N;

       count = 0;

**while** (value > 1){

              value = value/2;

              count++;

  }

1. Consider four programs—A, B, C, and D—that have the following performances:

**A -  O(log *n*) 11 seconds**

**B - O(*n*) 20 seconds**

**C - O(*n*2) 40 seconds**

**D - O(2*n*) 1.07151 ^ 605 seconds**

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.

1. 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](http://www.mscs.mu.edu/~praveen/Teaching/Fa17/DS1/Assignments/PrefixAverage.java). Implement both the algorithms 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.

|  |  |  |
| --- | --- | --- |
| **Size (N)** | **prefixAverage1** | **prefixAverage2** |
| **10** | **3ms** | **3ms** |
| **100** | **11ms** | **7ms** |
| **1000** | **49ms** | **45ms** |
| **10000** | **534ms** | **314ms** |
| **100000** | **8060ms** | **1974ms** |
| **1000000** | **--------** | **15,315ms** |