Homework 2 – Due Mar. 19th 23:59 KST

Instructions: Complete the report and turn it in before the due date. Any deviations from the instructed deliverable format will result in a deduction of grade. DO NOT COPY OTHER'S WORKS!

The answers to the following questions must be accompanied by appropriate justification. You will receive no credit if you do not show your work (drawing a graph does not count, except #4).

1. Compute the tightest order, in big-O notation, of the following functions and codes. In all cases, the variable to consider is 'n'. (5 pts x 4 = 20 pts) Some of these are not monotonic functions, but you should still be able to reason about the order of growth.

   A. $f(n) = 1 - 10000n^2 + 10^{-10}n^4$

   B. $f(n) = \frac{\sqrt{n}}{n} + \sqrt{n} + \sqrt{\log(n)}$

   C. A Java method that takes a sorted integer array of size 'n' as input and returns the minimum element.

   D.
```
for(int i = 0; i * i < n / 3; i++) {
    int[][] x = new int[2 * n][n];
    for(int j = 0; j < x.length; j++) {
        for(int k = 0; k < j; k++) {
            x[j][k] = i + 1;
        }
    }
}
```

2. Attach your code for HW1's whereIs(String) implementation. Based on YOUR code, analyze the big-O complexity of that method. The style of analysis should closely follow what we did in class for the insertion sort (20 pts). Please attach the code!!! We will NOT grade your work if you don't.

3. Open CSE214.java. Analyze the time complexities of the methods a() and b(). (10 pts)

4. Now execute CSE214.java multiple times with varying values for IN_SIZE to collect the running times of methods a() and b(). Draw a plot containing two curves, one for a() and one for b(), that depicts the actual running time (y-axis) in milliseconds vs. the size of the input (x-axis). What kind of trend can you find, and how does it compare to the theoretical analysis you gave in #3? (Don't just say 'They're the same' or 'They're different') Make sure to attach the plot as well. (10 pts)

5. Fill in the body of the following Java method. This method scans the given array 'arr' with 'item' and returns the number of elements in 'arr' that are less than or equal to 'item'. (e.g., If arr = {1, 2, 3, 6, 0, -1, 5} and item = 5, then count(arr, item) should return 6) The given array 'arr' is sorted in either ascending or descending order but it's unknown. Provide a complexity analysis on count() as we did in class, except now do it for two cases: best case and worst case (20 pts x 2 = 40 pts).

```
public static int count(int[] arr, int item) {
    // Return the # of elements that are <= item
}
```

Deliverables

- A single HW2.pdf file containing your name and SBU ID. The PDF file must contain all the answers to the problems above in a ***typed*** format. That is, don't hand-write your answers and attach the scanned version – it's very hard to discern poorly handwritten text. Do not turn in .java files for the coding problems: just include the code in your final PDF file.
- All codes you provide must be error-free, including syntax errors.