Data Structures and Algorithms
INFO 6205
Homework 1
Due: September 18, 2020

Put all your java work, compiled class files and documentation files into a zip file named Homework1.zip and submit it via the drop box on the blackboard before the END of due date. Put your name on all .java files. There will be a short quiz on this homework.

1. Why Algorithm Analysis is important both in terms of Running Times and Space complexity?

2. Formulate problem concerning exponents and logarithms such as 5^2 or $\log_4 16$, $\log_2 8$, $\log_3 27$. Draw the tree to solve it. Hopefully, the next time you look at a tree or plant you will see it a little bit more like a mathematician.

3. The Order of growth of an Algorithm is how long the time of execution depends on the length of the input array. Mathematically, show worst-case (upper-bound), average-case (tight-bound), best-case (lower-bound) of an algorithm. Explain clearly. What is asymptotic in order of growth?

4. Consider the following code:

A) What is the time-complexity of this algorithm?

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

B) What is the time-complexity of this algorithm?

```
int count = 0;
for (int i = N; i > 0; i /= 2)
   for (int j = 0; j < i; j++)
      count++;
```

5. Write code samples with worst-case running time of: (constant 1, logN, N, NlogN, N^2, N^3, 2^N). Mathematically, each example follows the following model, describe each case:

$$\binom{N}{3} = \frac{N(N-1)(N-2)}{3!} \sim \frac{1}{6}N^3$$

6. Estimate the running time (or memory) as a function of input size *N*. Explain as to why the results are the same for the following three examples.

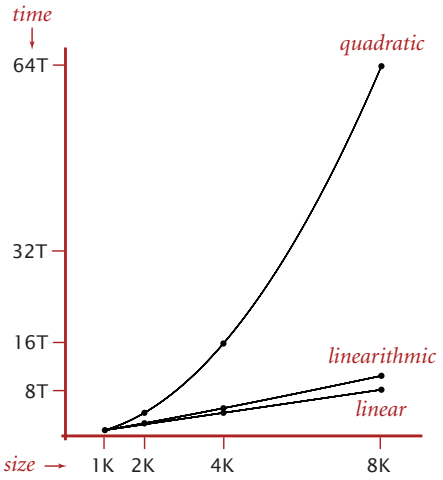
$\frac{1}{6} N^3 + 20N + 16 \sim \frac{1}{6} N^3$

$\frac{1}{6} N^3 + 100 N^{4/3} + 56 \sim \frac{1}{6} N^3$

$\frac{1}{6} N^3 - \frac{1}{2} N^2 + \frac{1}{3} N \sim \frac{1}{6} N^3$

7. Explain this graph

A)



B) Explain this data with various input sizes and measure running time, What is the graph looks like?

N	time (seconds) †
250	0
500	0
1,000	0.1
2,000	0.8
4,000	6.4
8,000	51.1
16,000	?

8. Explain as to why this is Brute-Force Algorithm;
What is the time complexity of this algorithm?

```
public class ThreeSum
{
   public static int count(int[] a)
   {
      int N = a.length;
      int count = 0;
      for (int i = 0; i < N; i++)
         for (int j = i+1; j < N; j++)
            for (int k = j+1; k < N; k++)
               if (a[i] + a[j] + a[k] == 0)
                  count++;
      return count;
    }
   public static void main(String[] args)
   {
      In in = new In(args[0]);
      int[] a = in.readAllInts();
      StdOut.println(count(a));
   }
```