# EE2331 Data Structure and Algorithm Homework 1

September 23 2023

**Due Time: 11:59PM, September 29, 2023**.
**No late work will be accepted. See the syllabus on Canvas about the details**

Problem 1 and 2 should be submitted in a pdf file. Please Name your Submitted Homework as: hw1-FamilyName-FirstName.pdf. For example, hw1-Chan-Keith.pdf. Follow the give order of the questions; don't change the order. Don't zip any file. Submit all files on Canvas using the link for hw1.

If you don't want to type, please make an effort to use neat, clean handwriting.

1. Consider the following functions in terms of order of magnitude. **First**, plot them in one or two figures (if one is too crowded). If needed, you can use log-scale coordinate for the Y-axis. You only need to show the final figures rather than the commands or methods. Pick the maximum N values so that the curves can show observable difference (e.g. you should not just plot the curves for just N $\leq$ 2). **Second**, order (list) them in terms of the growth, such that slow growth functions are listed before fast growth functions. That is, if f is listed before g, $f = O(g)$. If several functions have the same complexity, write them in the same line. Note that you don't need to submit the commands or codes for this problem. You can use Matlab, R, or a free web service to plot functions. (14 pts)

   N
   2/N
   37
   $N!$
   $N \log N$

$\log(N)$
$2^N$
$N \log^2(N)$
$N^{1.5}$
$N^2 \log(N)$
$N^3$
$N \log(N^2)$
$N^2$
$\sqrt{N}$

2. For each of the following three program fragments, give an analysis of the running time. **First**, give the total number of times of each statement being executed. **Then** give the summed/total time in Big-O notation using function with the least order of magnitude. Basically don't say the complexity is $O(N^9)$ if it is also $O(N^2)$. (21 pts = 3+5+4+9)

   (a)
   ```
   Sum = 0;
   for(i = 1; i ≤ N; i++)
       Sum = Sum + i;
   ```

   (b)
   ```
   Sum = 0;
   for(i = 1; i ≤ N^3; i++)
       for(j = 1; j ≤ N^2; j++)
           for(k = 1; k ≤ N; k++)
               Sum = Sum + 1;
   ```

   (c)
   ```
   Sum = 0;
   for(i = 1; i ≤ N; i++)
       for(j = 1; j ≤ i; j++)
           Sum = Sum + 1;
   ```

   (d)
   ```
   x = 100;
   y = 0;
   for(i = 1; i ≤ N; i = 2 * i)
       x = x + i;
       for(j = 1; j ≤ N; j++)
           if(x > y/j)
             y = y + i/j;
           else
             y = y - 1;
   ```

# EE2331 Data Structure and Algorithm Homework 1

September 23 2023

**Due Time: 11:59PM, September 29, 2023**.
**No late work will be accepted. See the syllabus on Canvas about the details**

Problem 1 and 2 should be submitted in a pdf file. Please Name your Submitted Homework as: hw1-FamilyName-FirstName.pdf. For example, hw1-Chan-Keith.pdf. Follow the give order of the questions; don't change the order. Don't zip any file. Submit all files on Canvas using the link for hw1.

If you don't want to type, please make an effort to use neat, clean handwriting.

1. Answer to the first problem.

   Answer:
   2/N
   37
   $\log(N)$
   $\sqrt{N}$
   $N$
   $N \log N$, $N \log(N^2)$
   $N \log^2(N)$
   $N^{1.5}$
   $N^2$
   $N^2 \log(N)$
   $N^3$
   $2^N$
   $N!$
   Note that $N \log(N^2)$ is $2N \log N$

2. For each of the following three program fragments, give an analysis of the running time. **First**, give the total number of times of each statement being executed. **Then** give the summed/total time in Big-O notation using function with the least order of magnitude. Basically don't say the complexity is $O(N^9)$ if it is also $O(N^2)$. (21 pts = 3+5+4+9)

(a)
```
Sum = 0;                  //1
for(i=1; i≤N; i++)        //N
    Sum = Sum + i;        //N
```

Answer: O(N).

(b)
```
Sum = 0;                          //1
for(i=1; i≤ N^3; i++)             //N^3
    for(j=1; j≤ N^2; j++)         //N^5
        for(k=1; k≤N; k++)        //N^6
            Sum = Sum + 1;        //N^6
```

Answer: O($N^6$)

(c)
```
Sum = 0;                      //1
for(i=1; i≤N; i++)            //N
    for(j=1; j≤i; j++)        //1+2+3+...+N
        Sum = Sum + 1;        //1+2+3+...+N
```

Answer: O($N^2$). Because $1 + 2 + 3 + ... + N = N(N + 1)/2$

(d)
```
x = 100;                      //1
y = 0;                        //1
for(i=1; i≤N; i=2*i)          //logN or 1+logN
    x = x + i;                //logN or 1+logN
    for(j=1; j≤N; j++)        //NlogN or N(1+logN)
        if(x > y/j)           //NlogN or N(1+logN)
            y = y + i/j;      //NlogN or N(1+logN)
        else                  //1 (actually a very small number
            y = y - 1;        //1 smaller than 1)
```

Answer: O($NlogN$).