# Data Structures and Algorithms

# INFO 6205

# Homework 2

# Due: September 19, 2019

Term: Qian Cai

NU ID:001389278

Put all your java, compiled class files and documentation files into a zip file named Homework2.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. The estimate running time (or memory) is a function of input size *N*. Explain as to why the results are the same for the following three examples.

⅙ *N* 3 + 20 *N* + 16 ~ ⅙ *N* 3

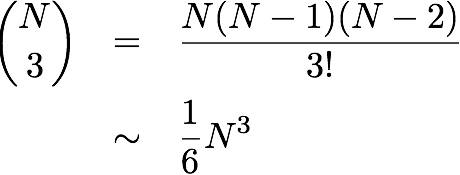
⅙ *N* 3 + 100 *N* 4/3  + 56 ~ ⅙ *N* 3

⅙ *N* 3 - ½ *N* 2+ ⅓ *N* ~ ⅙ *N* 3

**When N is large, lower-order terms are negligible.** **when N is small, we don't care. So we ignore lower-order terms. That is the reason that the results are the same for the following three examples.**

2. Write the Java code samples for the running times: constant 1, logN, N, NlogN, N^2,

N^3, 2^N. Mathematically, how do you describe each one of these examples in the form of

following equation?

**constant 1: 1**

**logN: 1/2+1/4+……1/N~logN**

**N: T(N)=2N ~ N**

**NlogN: T(N)=log1+log2+……logN ~ NlogN**

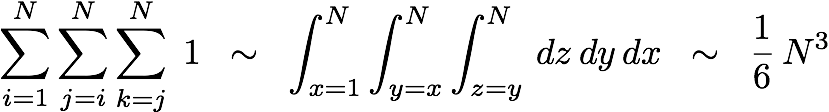
**N^2: T(N)=N+ ~**

**N^3: T(N)= N+ ~**

**2^N: T(N)=T(N-2)+T(N-1)~**

3. Write the code that results to following running time. The 3-Sum Triple loop has the following

running time estimate. Do Not prove the math. Just want explaining the math, what it

represents and why the result is 1/6 N^3

**Because the loop count of 3-Sum Triple loop is the count of choosing three number from N numbers. So the result is N(N-1)(N-2)/3!=1/6N^3.**

4. Human use Infix expression and computers use Postfix expression. You are to write a simple Calculator. There are three steps: a) Read Infix expression, b) Convert Infix expression to Postfix by hand, and c) Evaluate Postfix expression, d) Use the referenced c-program example and write it in Java code, compile and run with four Infix expression examples.

(1 + 2 \* (20 / 5 ))

(1 + 3 + ( ( 4 / 2 ) \* ( 8 \* 4 ) ))

(4 + 8) \* (6 - 5)/((3 - 2) \* (2 + 2))

(1 + 2 - 3)

**(1 + 2 \* (20 / 5 ))= 1 2 20 5/ \* + =9**

**(1 + 3 + ( ( 4 / 2 ) \* ( 8 \* 4 ) ))=1 3 4 2 / 8 4 \* \* + + =68**

**(4 + 8) \* (6 - 5)/((3 - 2) \* (2 + 2))=4 8 + 6 5 - \* 3 2 - 2 2 + \* / =3**

**(1 + 2 - 3)=1 2 + 3 - =0**

References:

<http://www.cs.nthu.edu.tw/~wkhon/ds/ds10/tutorial/tutorial2.pdf>

<https://www.includehelp.com/c/infix-to-postfix-conversion-using-stack-with-c-program.aspx>