# Data Structures and Algorithms

# INFO 6205

# Homework 2

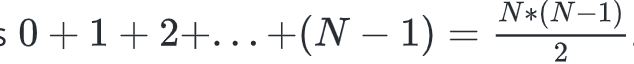
# Due: February 9, 2022

Put all your files and documentation files into a zip file named Homework2.zip and submit it via the drop box on Canvas before the END of due date.

NOTE: this homework is based on Lecture1Notes

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

2. 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?



3. Consider the following code:

A) Why the total count of this algorithm is: a)

and b) why time-complexity is O(N^2)?

int count = 0;

for (int i = 0; i < N; i++)

for (int j = 0; j < i; j++)

count++;

B) Why time-complexity of the following algorithm is O(N) and not O(N \* LogN)?

int count = 0;

for (int i = N; i > 0; i /= 2)

for (int j = 0; j < i; j++)

count++;

C) 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++;

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

int count = 0;

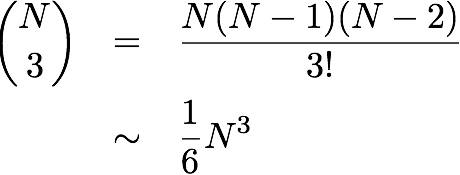
for (int i = N; i > 0; i /= 2)

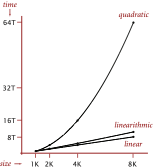
for (int j = 0; j < i; j++)

count++;

4. The worst-case running time of an Algorithm can be: (constant 1, logN, N, NlogN, N^2,

N^3, 2^N). Mathematically, each instance follows the following model, describe each case:



5. 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.

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

6. Explain this graph

7. 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));

}

}

9. Consider the following functions asymptotically:

A) true or false

B) draw the graph

C) explain Why true or false

a) 5n2-n+1 is Big O(n2)

b) 5n2-n-3 is Ω(n2)

c) 5n2 -n+1 is Big O(n3 )

d) 4n+1 is Θ(n)

e) 5n2-n+1 is Big O(n)

g) 4n+1 is Big O(n)

f) 4n-3 is Ω(n2)

h) 5n2-n-3 is Ω(n3)

i) 4n+1 is Big O(n2)

j) 4n+4 is Θ(n2)

k) 5n2-n+1 is Θ(n)

l) 4n+4 is Θ(n2)

m) 4n-3 is Ω(n)

n) n2+800 is Θ(n2)

o) 7n-2 = Θ(1)

10. Fill in the asymptotic relationship in table below:

