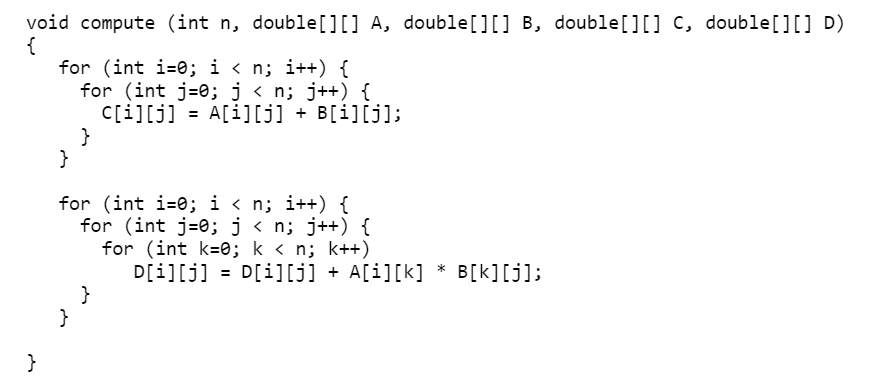
Timothy Zwart

9/14/2018

CMPT\_435\_111\_18F

Assignment 2

1. Analyze the following code and provide a "Big-O" estimate of its running time in terms of n. Explain your analysis.



For(int i=0; i< n; i++){ 🡪n

For(int j=0; j< n; j++){ 🡪n

C[i][j]=A[i][j]+B[i][j]🡪1

}

}🡪 n\*n\*1= n^2

For(int i=0; i< n; i++){🡪n

For(int j=0; j< n; j++){🡪n

For(int k=0; k< n; k++){🡪n

D[i][j]=D[i][j]+A[i][k]\*B[k][j]🡪1

}

}

}🡪n\*n\*n\*1=n^3

N^2 + N^3 🡪 Big O = O(N^3) because it is the highest power

1. Analyze the following code and provide a "Big-O" estimate of its running time in terms of n. Explain your analysis.

int j = 1, i = 0;

while (i < n)

{

i = i + j;

j++;

}

The loop will run the same amount of times as to whatever variable j becomes -1.

Ex n=36

1=0,j=1

First: i=1,j=2

Second: i=3, j=3

Third: i=6, j=4;

Fourth: i=10,j=5;

Fifth : i=15,j=6;

Sixth : i=21,j=7;

Seventh : i=28,j=8;

Eighth (last time it loops because now n and I are equal): i=36,j=9;

O(j)🡪O(N)

Big O is O(N) because of linear complexity.