

NATNAEL MENGISTU

Assignment 2

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

```
void compute (int n, double[][] A, double[][] B, double[][] C, double[][] D)
{
    for (int i=0; i < n; i++) {
        for (int j=0; j < n; j++) {
            C[i][j] = A[i][j] + B[i][j];
        }
    }

    for (int i=0; i < n; i++) {
        for (int j=0; j < n; j++) {
            for (int k=0; k < n; k++)
                D[i][j] = D[i][j] + A[i][k] * B[k][j];
        }
    }
}
```

Answer

- We ignore loop head on 3rd, 4th, 9th, 10th and 11th line while we count the primitive operations on it
so from line #5 we find $C[i][j] = A[i][j] + B[i][j]$; we get //2 primitive equations
#12 we find $D[i][j] = D[i][j] + A[i][k] * B[k][j]$; //3 primitive equations

while doing iteration of the first and second for loops, we ran the first loop n times and the second loop runs for every single time the first one runs so we have a $O(n^2)$ notation.

After these loop is completed we have 3 nested for loops which iterate N times which we will have $O(n^3)$ notation.

Since other lines are linear iteration we get a **$O(n^3)$** estimate run time.

2. 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++;
}
```

Hint: The loop variable 'i' is incremented by 1, 2, 3, 4, ... until i becomes greater than or equal to n.

Answer

#1 //2

we ignore the loop head on the second line

#4 //2

while doing Iteration for the given pseudocode keeps running n times until $i \geq n$
so the following loop has running time notation of **$O(n)$**