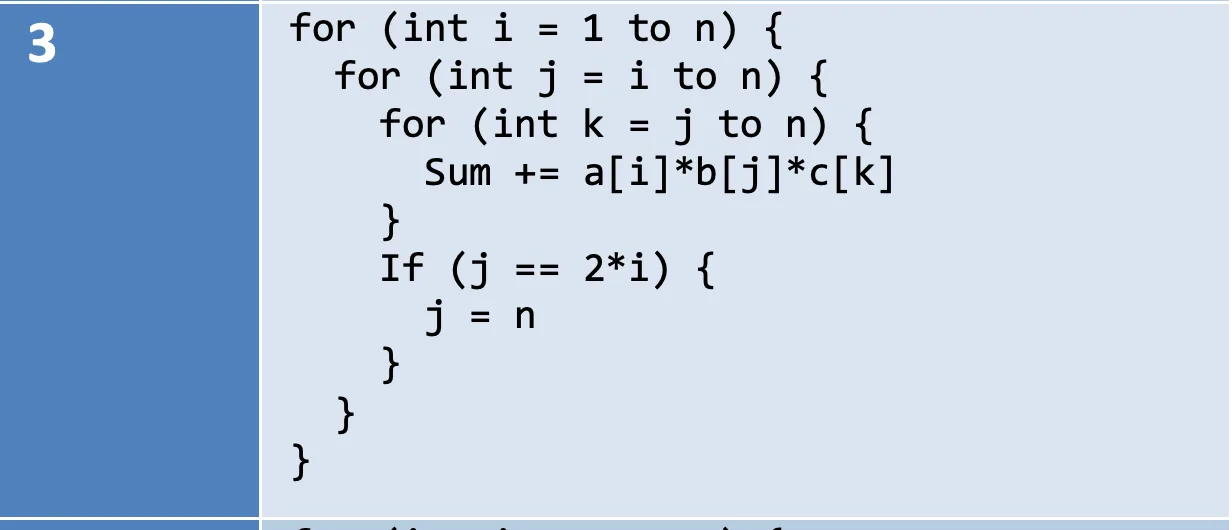
Project 1

Alekya Sai Laxmi Kowta

# Problem Statement

Analyze and write the time complexity of the following program:



# Theoretical Analysis

The theoretical time complexity of the program is O(n3).

The program consists of 3 nested loops.

Outer loop

Middle loop

Inner loop

The if statement does cause the loop to terminate early, but this occurs only if .

For the remaining iterations where , the condition is never met, and the loops do not break.

The number of operations in this worst-case scenario dominates the time complexity, hence the if statement would not dramatically change the complexity of the program.

The total number of operations can be broken down into 2 parts:

* One where if condition is met
* One where if condition is not met

If :

The condition is never met as and j loop range is

The number of operations is a triple summation:

If :

j loop terminates as soon as .

The number of operations is:

These number of operations are lesser than the first case, but it still will amount to a cubic function of n.

Therefore, the time complexity is O(n3)

# Experimental Analysis

## Program Listing

N Values used in the program:

int[] nValues = {2, 10, 50, 75, 100, 200, 225, 400, 450, 500, 600, 750, 800, 1200, 1600, 1800, 2200, 2400, 4000, 6000, 8000};

## Data Normalization Notes

Theoretical time complexities (such as n3) do not have physical units, while experimental times are measured in nanoseconds. To visually and quantitatively compare them on the same plot, we adjust (or normalize) the theoretical values with a scaling constant.

Using n = 2400 as the reference point, we calculate the scaling constant by dividing the experimental time at the reference point by the corresponding theoretical value.

In this analysis, the scaling constant is approximately

Then, we apply the scaling constant to all the theoretical values.

## Output Numerical Data

|  |  |  |  |
| --- | --- | --- | --- |
| n | experimental\_time\_ns | theoretical\_time | adjusted\_theoretical\_time |
| 2 | 99041 | 8 | 0.2309027777777778 |
| 10 | 9750 | 1000 | 28.862847222222225 |
| 50 | 144875 | 125000 | 3607.855902777778 |
| 100 | 747375 | 1000000 | 28862.847222222223 |
| 200 | 1333875 | 8000000 | 230902.77777777778 |
| 400 | 2228625 | 64000000 | 1847222.2222222222 |
| 800 | 15479125 | 512000000 | 1.4777777777777778E7 |
| 1600 | 89833708 | 4096000000 | 1.1822222222222222E8 |
| 2400 | 378360541 | 13824000000 | 3.99E8 |  |
| 6000 | 6093974833 | 2.16E+11 | 6.234375E9 |  |
| 8000 | 14500381708 | 5.12E+11 | 1.4777777777777779E10 |  |

## Graph

Graph made in Python using matplotlib and numpy.

Experimental Vs Theoretical Time Graph Plot


## Graph Observations

The above plot, “Experimental vs Adjusted Theoretical Time”, effectively compares the time taken by the code and the theoretical time. It is observed that the n values overlap closely across most of the tested values of n, indicating positive results for the theoretical analysis provided.

As n grows larger, both curves show rapid increase as expected, and differences are negligible as n increases.

The addition of more data points has helped the plot appear smoother and has clarified the strong correlation between the measured and theoretical times for the input values n.

# Conclusions

The observed experimental results confirm that the complexity of the given algorithm is **O(n3).**

# GitHub

<https://github.com/AlekyaKowta/CSCI_6212_12_Project_1>

Readme: <https://github.com/AlekyaKowta/CSCI_6212_12_Project_1/blob/main/README.md>