Intro to Algorithm

Report

109550198

* **Environment (Os, compiler version, IDE)**

To implement this program, which is using a scale, which can only test which side (left or right) is heavier to find this fake coin. The OS used for this program is MacOS, the compiler version is apple clang version 14.0.0, and the IDE is visual studio code for mac.

* + - How to run your program

To run the program, I used the terminal of vs code because I was already in the file directory and typed *g++ 109550198.cpp* . After I clicked enter and typed *./a.out <input.txt* because the program uses standard input and output, I pressed enter. The terminal shows me the result

Here is the screenshot:

A screenshot of a computer

Description automatically generated

* Results
  + - * + Method or solutions

The fake coin problem was solved recursively. To identify the fake coin from the given collection of N coins, the Divide and conquer algorithm is the best one to find the fake coin with a lesser number of comparisons. The algorithm is given different steps to find the solution for the given problem.

**Explanations**

* Divide the N coins into 3 equal groups and 1 remainder

Group1=N/3

Group2=N/3

Group 3=N/3

Group4=remainder

* Consider the function findFakeCoin ().

It is splitting the group variable into four groups (3 equal groups and 1 remainder ). After that, the getWeight**()** function is being called for each group. A recursive call is being made with any one of the groups.

Here is the algorithm:

if group1 equal to group2:

if the remaining group is empty:

if the length of group 3 equal to 2

Take the first element in group 1

return find Fake coin (group3)

else:

if group3 equal to group1 (or group2):

if the length of the remainder group equal to 2

Take the first element in group 2

return findFakecoin (remainder group)

else:

if the length of group 3 equal to 2

take the first element in group 1

return findFake(group3)

else if group1 not equal to group2:

if group3 equal to group1:

if the length of group2 equal to 2

take the first element in group 3

return findFakecoin(group2)

else:

if the length of group2 equal to 2

take the first element in group 3

return findFake(group1)

* Analyze the running time.

Step 1: Here are some functions in the code. Each function has its own time complexity which is used in the other functions as nested one. Firstly, the getWeight**()** function is defined as -

int getWeight(vector < int > group) {

int sum = 0;

for (int i = 0; i < group.size(); i++) {

sum += group[i];

}

return sum;

}

This function is hence running from 0 to the size of the group which comes under the linear time complexity. Let us say O(N).

Step 2:

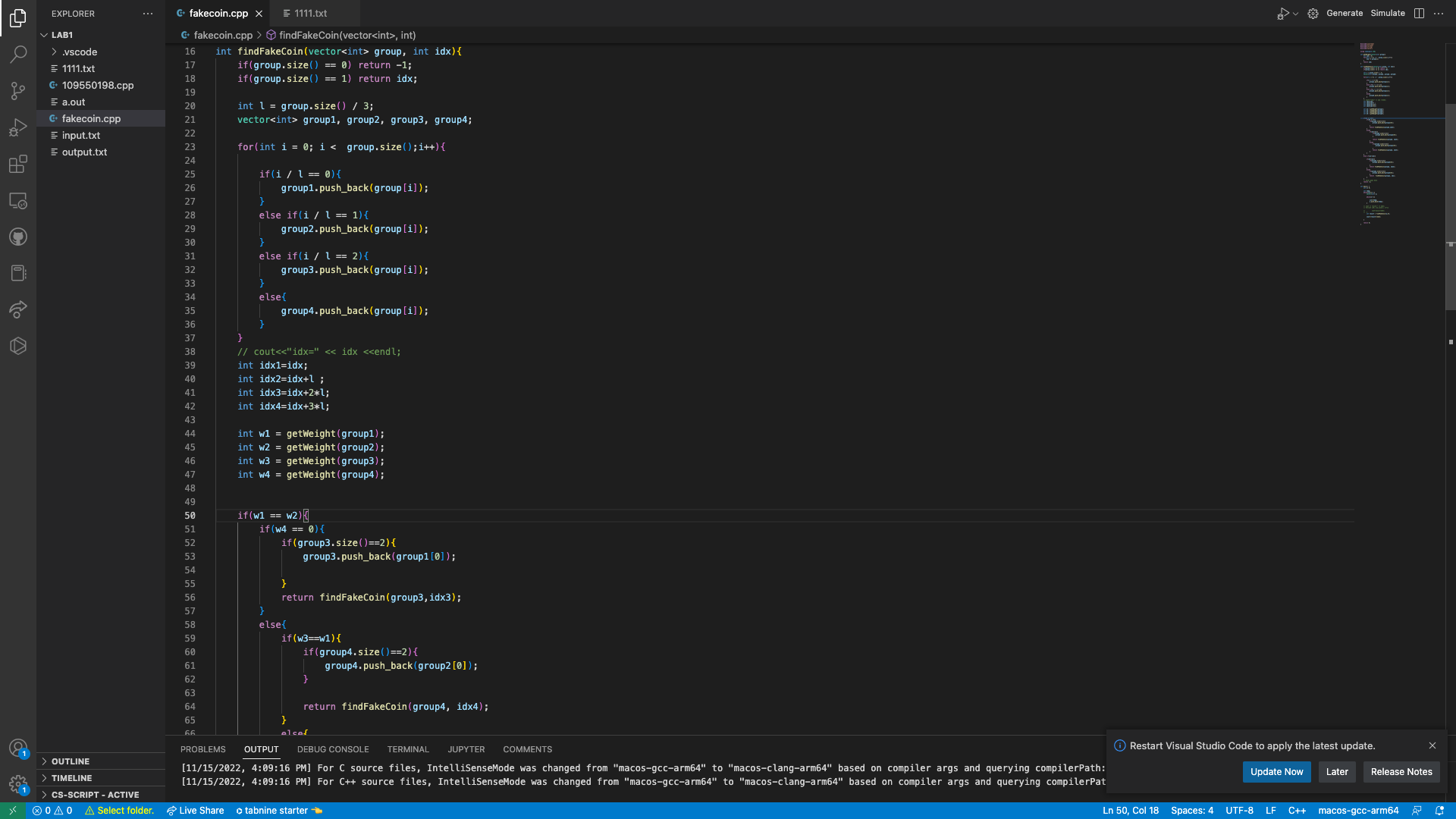
The time complexity of this code is O(1).

A screenshot of a computer

Description automatically generated with medium confidence

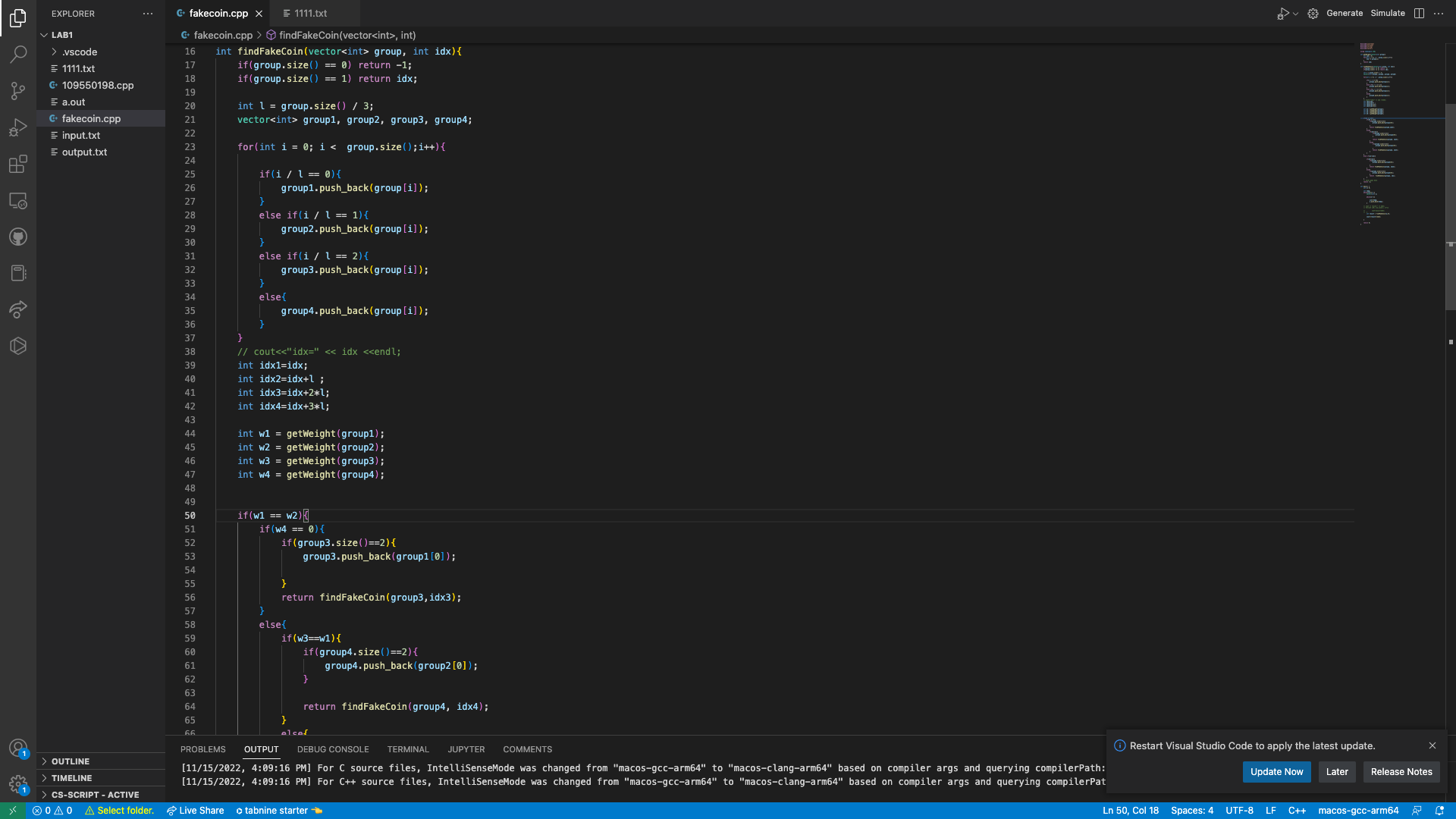
Step 3:

The time complexity of this code is O(N)



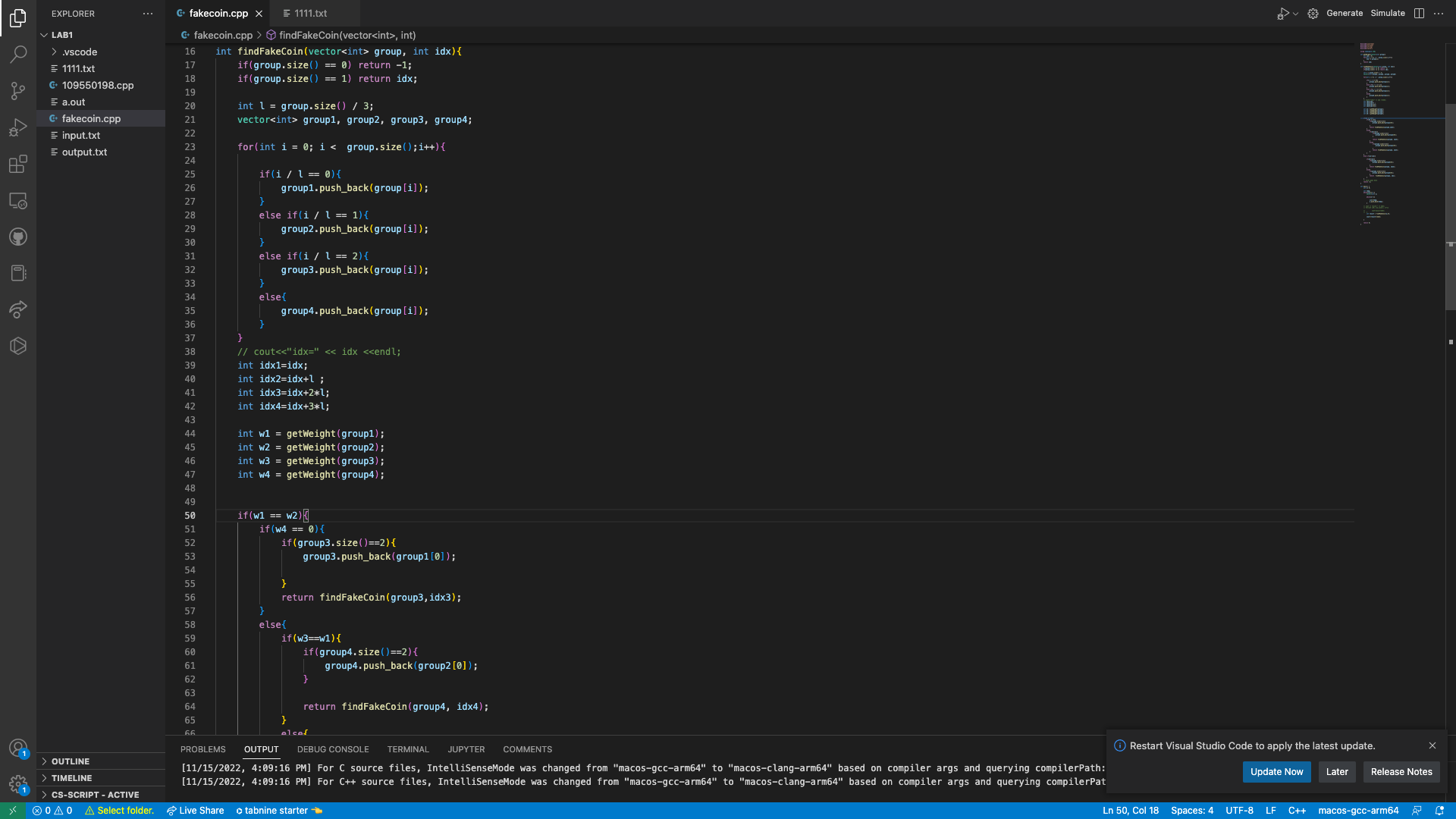
Step 4:c

The time complexity of this code is O(1)

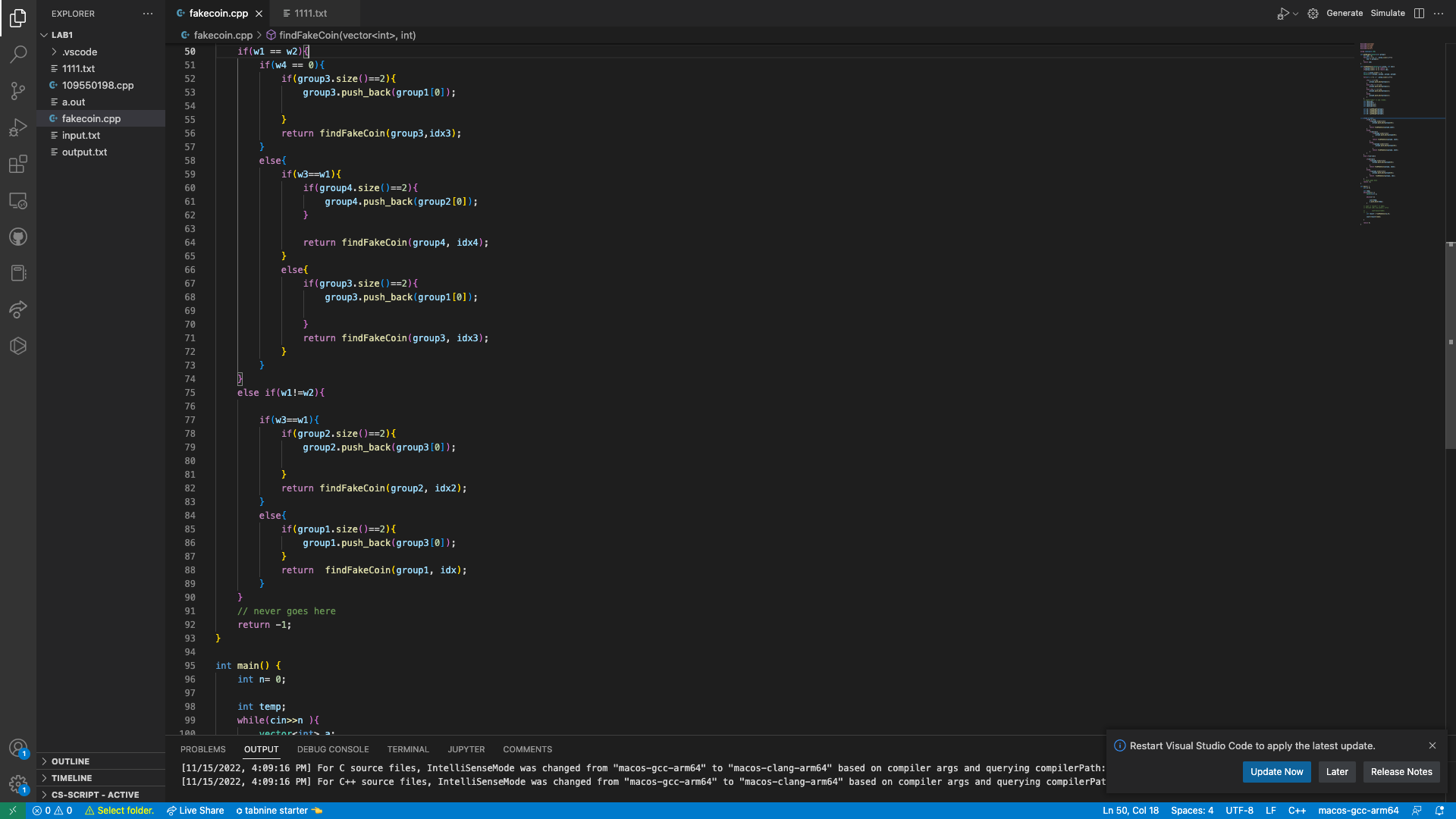


Step 5:

The time complexity of this code is O(N)



Step 6:

The time complexity of this code is T(N/3)

* Explanation

The time complexity of the findFakeCoin() function from step 1 to step 6:

T(N)=O(N)+O(1)+O(N)+O(1)+O(N)+T(N/3)

Simplification :

T(N)= T(N/3)+O(N)

Final answer:

Applying the master method, the time complexity is O(N).

* Anything you want to share

I made a mistake in analyzing the time complexity in the previous report. In this report, I determined the time complexity by analyzing the program's statements ( I go line by line)