Consider the following algorithm.
Algorithm 1 function

Mystery(n)
r ← 0
for i ← 1 to n − 1 do
for j ← i + 1 to n do
for k ← 1 to j do
r ← r + 1
end for
end for
end for
return r

(a) What is the value returned by the function Mystery? Express your answer as a function of n and give the closed form.
(b) Using O() notation, give the worst-case running time of the function Mystery.

This is a very interesting problem, as the name says Mystery.

If you write the same problem in a simple programming language, then it looks something like

```
#include <iostream>

using namespace std;


int main() {


    int n;

    cin >> n;

    int r = 0;
```

```
    for (int i = 1; i <= n-1; i++){
        for (int j = i+1; j <= n; j++){
            for (int k = 1; k <= j; k++){
                r++;
            }
        }
    }
    cout << r << endl;
    return 0;
}
```

So it basically has 3 for loops (in nested fashion)

Let's take an example to feel the problem

Let n = 5

The outermost 'for' loop goes from i = 1 to i = 4

And for each for these 'i', the inner 'for' loops work,

Say for i = 1

j goes from i+1 to n, that is, 2 to 5

and for each value of j, innermost loop goes from k = 1 to k = j and increments the value of 'r' in each iteration.

So for j = 2

k goes from 1 to 2.

Finally, value of r is returned.

So, r gets incremented two times for j =2, three times for j = 3, …, five times for j = 5

In similar manner for other values of i

So final value of r = (2 + 3 + 4 + 5) {corresponding to i = 1} + (3 + 4 + 5) {corresponding to i = 2} + (4 + 5) {corresponding to i = 3}+ (5) {corresponding to i = 4}

= 1 * 2 + 2 *3 + 3*4 + 4*5

= 40

GENERALIZATION {EXPRESSION FOR VALUE of r}:

IF we write in terms of n;

Value of r = 1 * 2 + 2* 3 + 3 * 4 + ….. + (n-1) * n

Or write r = $\sum^{n-1}_{i=1}$ (i) * (i+1) [That is summation from i = 1 to i = n-1]

TIME COMPLEXITY:

Since there are 3 for loops, so in worst case,

Time complexity can go as bad as O (n^3).

BONUS: [This is just for extra knowledge, not asked in question]

You can return the same value of r by doing an optimization

Say

Intialize r = 0

For (int i = 1; i <= n-1; i++){

    r = r + (i)* (i+1)

}

Return r

This will also return same value of r, but with complexity just O (n), linear time.

I hope it helps ☺