Problem - 01

Name of the Problem : Find the Complexity of a Loop.

Tools : Visual Stdio Code, GNU G++ Compiler, C++.

Algorithm1.1:

1. Repeat for K = 1 to n by 1.
2. Write: K (End of Step 1 loop).
3. Exit.

Source Code :

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(void){

int n;
cout << "Size of the loop : ";
cin >> n;

auto start = chrono::high_resolution_clock::now();

for (int K = 1; K <= n; K++){
    cout << K << ' ';
}
cout << "\n\n";

auto end = chrono::high_resolution_clock::now();

chrono::duration<double> duration = end - start;

cout << "Execution time: " << duration.count() << " seconds\n";
}
```

Table :

n	f(n) (from Program, Count Statement)	cg(n)(Theoretical)
100	0.000107 seconds	
200	0.000125 seconds	

Graph :

Comments :

Problem - 02

Name of the Problem : Find the Complexity of the following Program.

Tools : Visual Stdio Code, GNU G++ Compiler, C++.

Algorithm1.2:

1. Repeat for K = 1 to n by 1.
2. Repeat for L = 1 to n by 1.
3. Write: L (End of Step 2 loop).
4. Write: K (End of Step 1 loop).
5. Exit.

```
#include <bits/stdc++.h>
using namespace std;

int main(void){
int n;
cout << "Size of the loop : ";
cin >> n;

auto start = chrono::high_resolution_clock::now();

for (int K = 1; K <= n; K++){
    for (int L = 1; L <= n; L++){
        cout << L << ' ';
```

```
    }
    cout << "\nK = " << K << '\n';
}
cout << "\n\n";

auto end = chrono::high_resolution_clock::now();
chrono::duration<double> duration = end - start;
cout << "Execution time: " << duration.count() << " seconds\n";
}
```

Table :

n	f(n) (from Program, Count Statement)	cg(n)(Theoretical)
100	0.002475 seconds	
200	0.028762 seconds	

Graph :

Comments :

Problem - 03

Name of the Problem : Find the Complexity of the elementary Sort algorithm.

Tools : Visual Stdio Code, GNU G++ Compiler, C++.

Algorithm1.3:
Given a nonempty array A with n numerical values. This algorithm sorts the values.

1. Repeat for i = 2 to n by 1
2. Repeat for k = i to 1 by -1
3. If A[k]<A[k-1] then:

Swap (A[k], A[k-1]) [End of If Structure] [End of Step 2 loop] [End of Step 1 loop]

4. Exit