Timus Online Judge

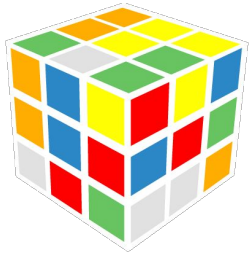
1878.

Rubinchik's Cube

Time limit: 0.5 second
Memory limit: 64 MB
Difficulty: 192

CS3201 | Algorithm Design | 1-2021

Team



Minn Moe Yan Oo U6017540

Vibolrottnak Seng U6118173

Content

1 **Problem Description**

Explanation of the Problem

2 **Analysis**

Understanding the Input and Output

3 **Implementation**

Problem Approach and Code Explanation

4 **Conclusion**

Submission Result

1

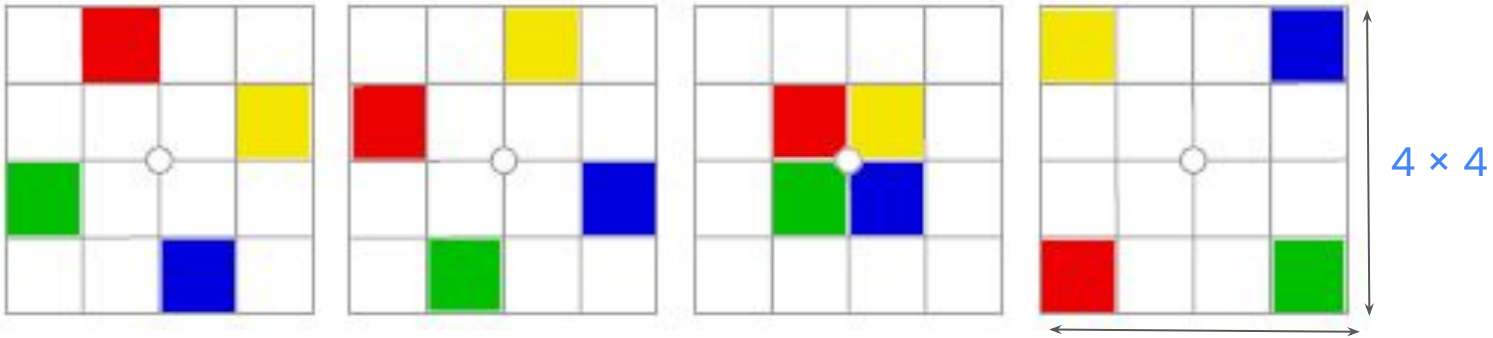
Problem Description

Explanation of the Problem

Problem Description

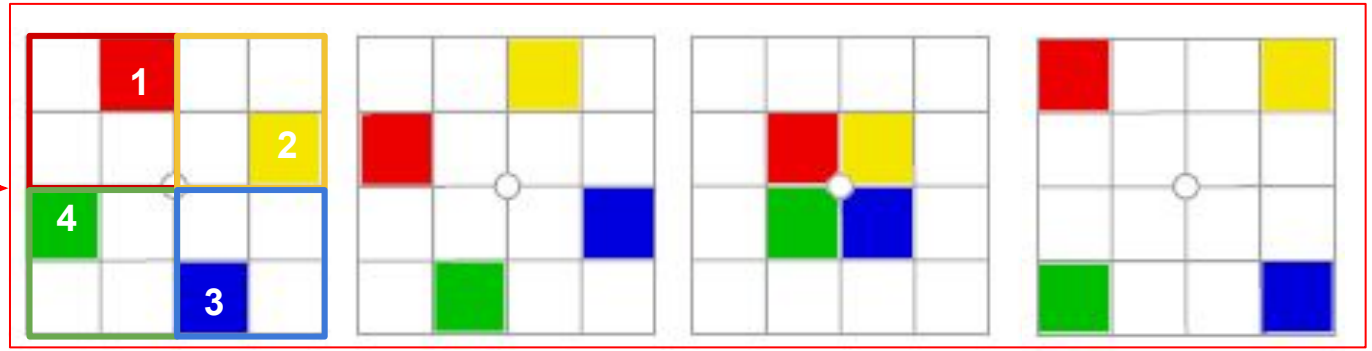
- This rubik's cube consists of four layers of size 4 × 4.

  The layers look as follows.



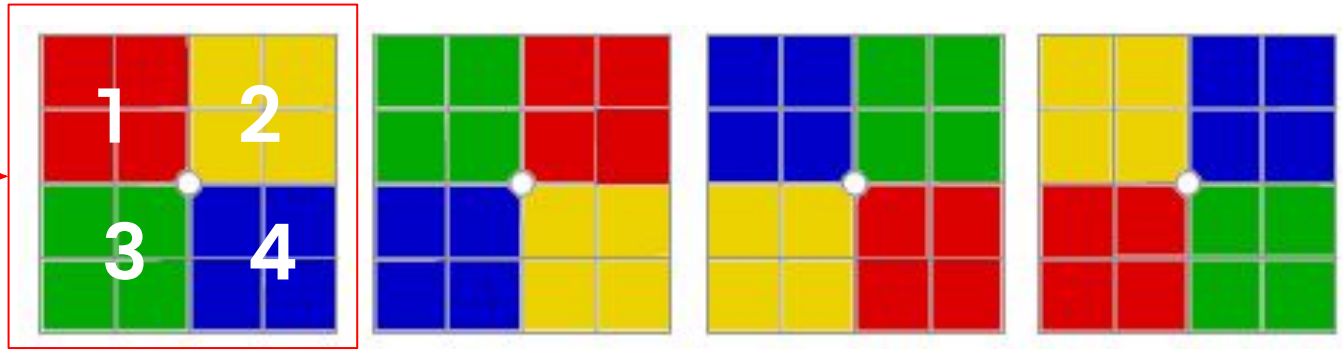
- Each layer is made of glass and is absolutely transparent.
- The upper face of each layer is divided into 16 equal squares.

- Four of the squares of each layers are painted red, yellow, blue, or green in their respective positions



Look on
Separate layers

- The cube is solved if it is in one of the following states (top view).



Solved States!

Input

You are given four lines, each containing four integers, which describe the current state of the cube (top view). The numbers are in the range from 1 to 4; they correspond to the colors of the cube's squares. Different numbers denote different colors.

Output

Output the minimum number of turns needed for solving the cube.

Sample

Input	Output
2 1 2 3 1 1 2 2 4 4 3 3 1 4 3 4	1

2

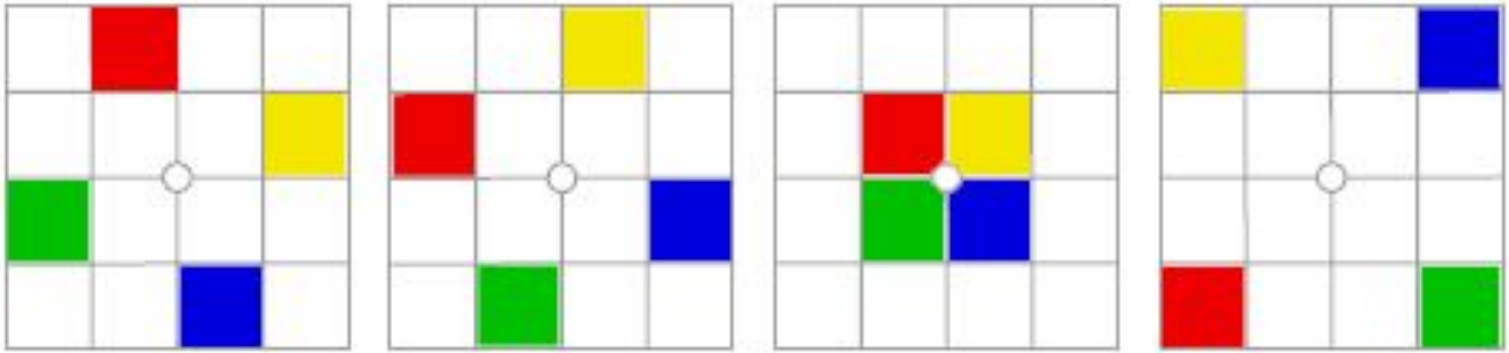
Analysis

Understanding the Input and Output

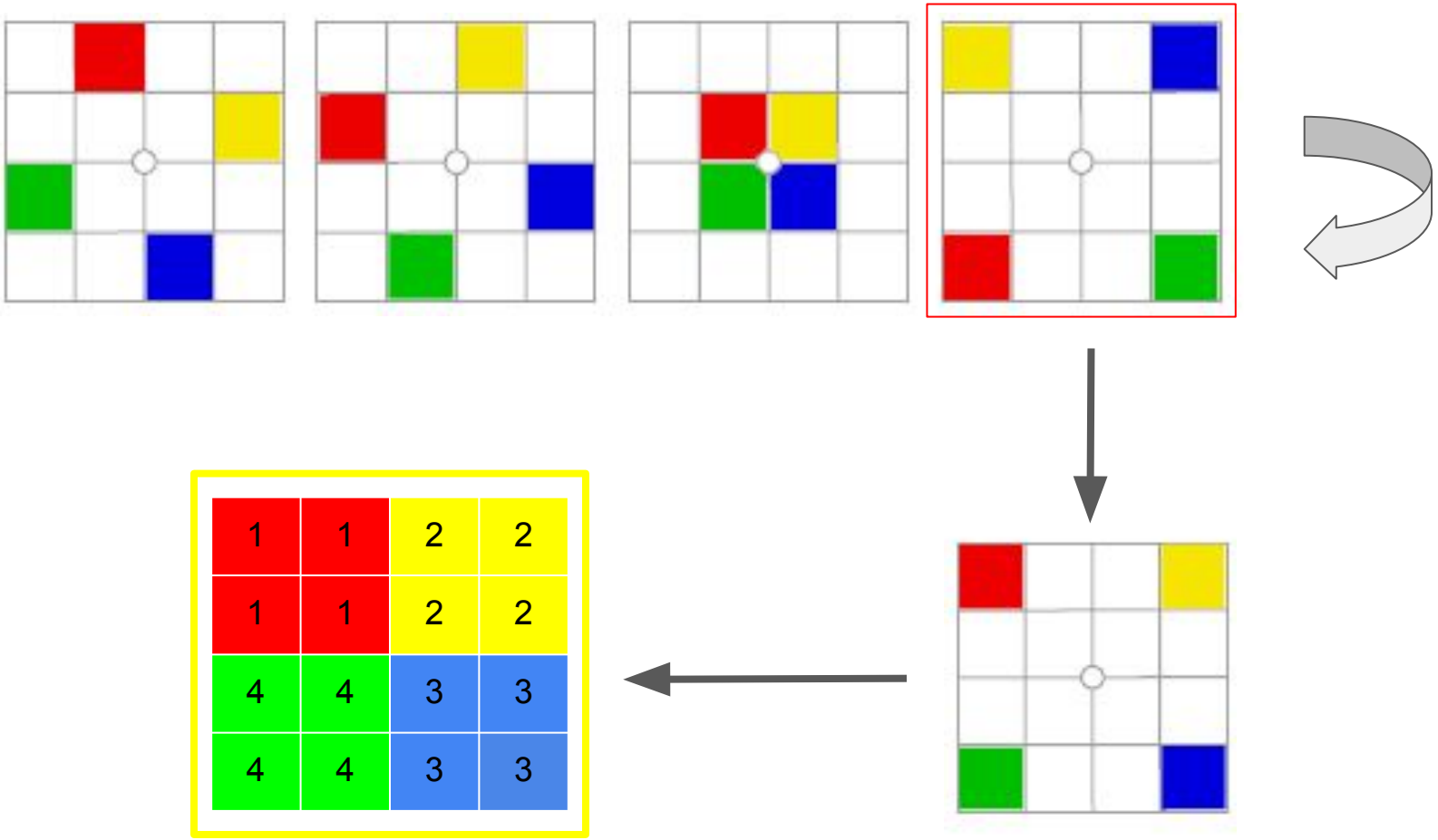
2	1	2	3
1	1	2	2
4	4	3	3
1	4	3	4

Input	Output
2 1 2 3 1 1 2 2 4 4 3 3 1 4 3 4	1

View when put on top of each other



View of the separate layers

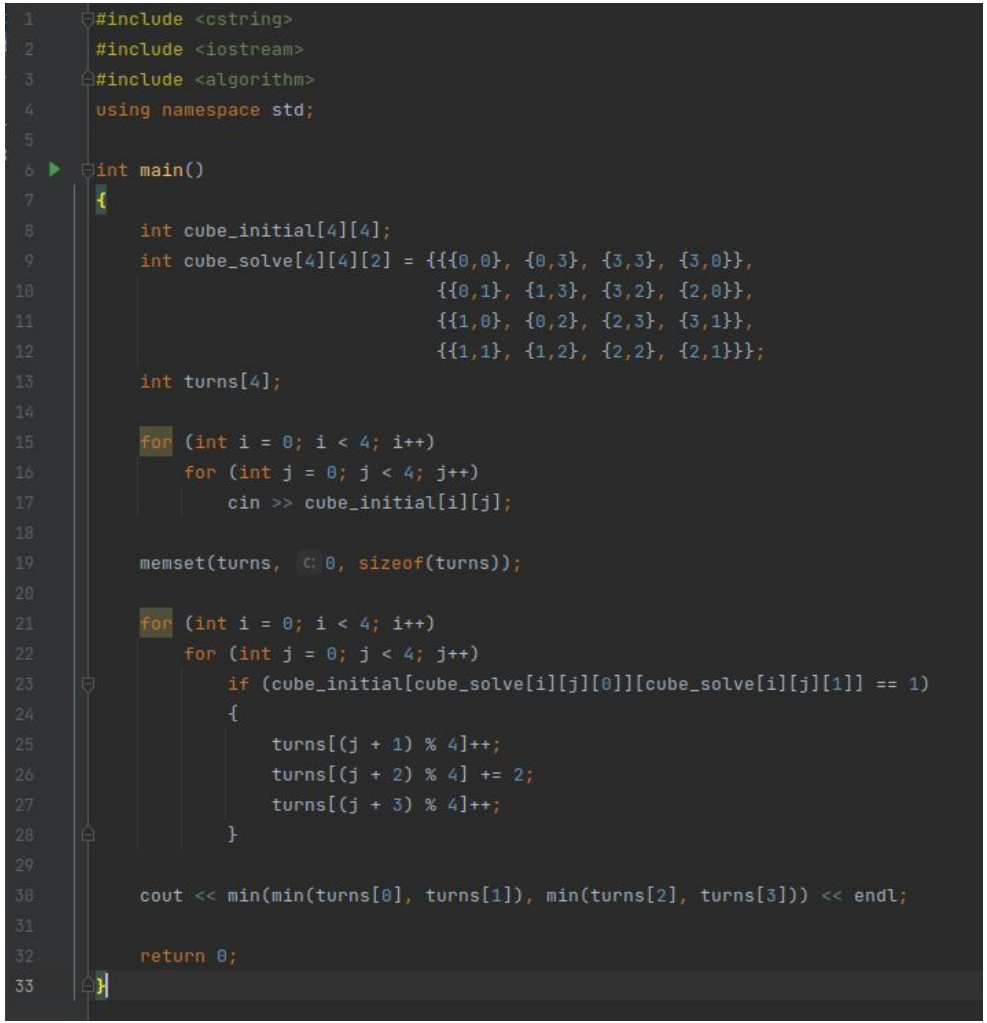

1 1 2 2
1 1 2 2
4 4 3 3
4 4 3 3

3

Implementation

Problem Approach and Code Explanation

C++



```cpp
#include <cstring>
#include <iostream>
#include <algorithm>
using namespace std;

int main()
{
    int cube_initial[4][4];
    int cube_solve[4][4][2] = {{{0,0}, {0,3}, {3,3}, {3,0}},
                               {{0,1}, {1,3}, {3,2}, {2,0}},
                               {{1,0}, {0,2}, {2,3}, {3,1}},
                               {{1,1}, {1,2}, {2,2}, {2,1}}};
    int turns[4];

    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4; j++)
            cin >> cube_initial[i][j];

    memset(turns, 0, sizeof(turns));

    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4; j++)
            if (cube_initial[cube_solve[i][j][0]][cube_solve[i][j][1]] == 1)
            {
                turns[(j + 1) % 4]++;
                turns[(j + 2) % 4] += 2;
                turns[(j + 3) % 4]++;
            }

    cout << min(min(turns[0], turns[1]), min(turns[2], turns[3])) << endl;

    return 0;
}
```

4

Conclusion

Submission Result

Submission

9465997	00:16:03 22 Sep 2021	minnmoeyanoo	1878. Rubinchik's Cube	Visual C++ 2019 x64	Accepted		0.015	260 KB

Resources

- https://acm.timus.ru/problem.aspx?space=1&num=1878
- https://acm.timus.ru/forum/?space=1&num=1878
- https://github.com/anthonymirand/CodingPractice/blob/master/1878%20-%20Rubinchik's%20Cube.cpp