1st LGE Code Jam 2019 - Problem B (English)

Created by 허지영 jiyoung.huh, last modified by 박은영 eunyoung.park on 2019/05/14

[Array Manipulation]

If you want to solve this problem with the Brute Force algorithm to sequentially process all cases for a given condition, the time complexity of $O(MN^2)$ is required to process M operations on all the elements of the two-dimensional array of N x N.

In this problem, since the sizes of N and M are 1 <= N, M <= 1000 and the time complexity of the algorithm is $O(MN^2) = 10^9$, the total execution time for test case T <= 10 is $O(TMN^2) = 10^{10}$ and the problem can not be solved within a given time (2 seconds).

However, the output required by the problem does not print all the elements of the two-dimensional array after M operations, but prints only the sum of each row and column.

If you think a bit more, instead of applying M operations to elements of rows from r1 to r2 and columns from c1 ~ c2,we can consider an algorithm that updates only the sum of each row and column using two one-dimensional arrays.

In this case, the time complexity is O(MN) because it computes only the sum of rows from r1 to r2 and columns from c1 ~ c2.

Total execution time for test case T <= 10 is $O(TMN) = 10^7$ and the problem can be solved within the time (2 seconds).

Problem B

```
#include <cstdio>
#include <cstdlib>
#include <cstring>

using namespace std;

#define MAX_N   1001
int N, M;
int R[MAX_N];
int C[MAX_N];

void solve()
{
    int x, v;
    int r1, r2, c1, c2;

    for(int i=0; i<M; i++) {
        scanf("%d %d %d %d %d", &r1, &c1, &r2, &c2, &v);
        r1--; r2--; c1--; c2--;
        x = (r2-r1+1) * v;
        for(int j=c1; j<=c2; j++) C[j] += x;
        x = (c2-c1+1) * v;
        for(int j=r1; j<=r2; j++) R[j] += x;
    }

    for(int i=0; i<N; i++) printf("%d ", R[i]); printf("\n");
    for(int i=0; i<N; i++) printf("%d ", C[i]); printf("\n");
}

int main(int argc, char* argv[])
{
    int T;
    int x;

    scanf("%d", &T);
    while(T-->0) {
        memset(R, 0, sizeof(R));
        memset(C, 0, sizeof(C));
        scanf("%d %d", &N, &M);
        for(int i=0; i<N; i++) {
            for(int j=0; j<N; j++) {
                scanf("%d", &x);
                R[i] += x;
                C[j] += x;
            }
        }
        solve();
    }


    return 0;
}
```