

구간 합 구하기 5

문제

$N \times N$ 개의 수가 $N \times N$ 크기의 표에 채워져 있다. $(x1, y1)$ 부터 $(x2, y2)$ 까지 합을 구하는 프로그램을 작성하시오. (x, y) 는 x 행 y 열을 의미한다.

예를 들어, $N = 4$ 이고, 표가 아래와 같이 채워져 있는 경우를 살펴보자.

1	2	3	4
2	3	4	5
3	4	5	6
4	5	6	7

여기서 $(2, 2)$ 부터 $(3, 4)$ 까지 합을 구하면 $3+4+5+4+5+6 = 27$ 이고, $(4, 4)$ 부터 $(4, 4)$ 까지 합을 구하면 7이다.

표에 채워져 있는 수와 합을 구하는 연산이 주어졌을 때, 이를 처리하는 프로그램을 작성하시오.

입력

첫째 줄에 표의 크기 N 과 합을 구해야 하는 횟수 M 이 주어진다. ($1 \leq N \leq 1024$, $1 \leq M \leq 100,000$) 둘째 줄부터 N 개의 줄에는 표에 채워져 있는 수가 1행부터 차례대로 주어진다. 다음 M 개의 줄에는 네 개의 정수 $x1, y1, x2, y2$ 가 주어지며, $(x1, y1)$ 부터 $(x2, y2)$ 의 합을 구해 출력해야 한다. 표에 채워져 있는 수는 1,000보다 작거나 같은 자연수이다. ($x1 \leq x2, y1 \leq y2$)

출력

총 M 줄에 걸쳐 $(x1, y1)$ 부터 $(x2, y2)$ 까지 합을 구해 출력한다.

예제 입력 1

```
4 3
1 2 3 4
2 3 4 5
3 4 5 6
4 5 6 7
2 2 3 4
3 4 3 4
1 1 4 4
```

예제 출력 1

```
27
6
64
```

예제 입력 2

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

예제 출력 2

```
1
2
3
4
```

문제 해설

11659번: 구간 합 구하기 4

BAEKJOON>
ONLINE JUDGE

</> <https://www.acmicpc.net/problem/11659>

이 문제의 이차원 배열 버전?이다.

이 문제는 그림으로 그려보면 쉽다.

구간합을 구하는 법

1	3	6	10
3	8	15	24
6	15	27	42
10	24	42	64
sum[i][j-1]			
1	3	6	10
3	8	15	24
6	15	27	42
10	24	42	64
sum[i-1][j]			
1	3	6	10
3	8	15	24
6	15	27	42
10	24	42	64
sum[i-1][j-1]			
1	3	6	10
3	8	15	24
6	15	27	42
10	24	42	64
arr[i][j]			

1	3	6	10
3	8	15	24
6	15	27	42
10	24	42	64
sum[y_1-1][x_2]			
1	3	6	10
3	8	15	24
6	15	27	42
10	24	42	64
sum[y_2][x_1-1]			
1	3	6	10
3	8	15	24
6	15	27	42
10	24	42	64
sum[y_2][x_2]			
1	3	6	10
3	8	15	24
6	15	27	42
10	24	42	64
sum[x_1-1][y_1-1]			

1행과 1열은 앞선 문제의 1차원 구간합처럼 구하면 된다.

근데 2행 2열부터 어떻게 구간합을 구할 것인가는 (왼쪽 사진)

빨간색 + 주황색 - 노란색 + 초록색 = 초록색 까지의 구간합

식으로 표현하면

sum[i][j-1] + sum[i-1][j] - sum[i-1][j-1] + arr[i][j]이다.

노란색을 빼줘야 하는 이유는 빨간색과 주황색의 공통부분 이기 때문에 2번 더해지기 때문이다.

구간합을 구하고 나서는?

예제의 (2,2)~(3,4)까지의 구간합을 구해보자(오른쪽 사진)

노란색 - 빨간색 - 주황색 + 초록색 = (2,2)~(3,4)의 구간합

식으로 표현하면

sum[y_2][x_2] - sum[y_2][x_1-1] - sum[y_1-1][x_2] + sum[x_1-1][y_1-1]

초록색 구간을 다시 더해주는 이유는 빨간색 구간과 주황색 구간에서 두 번 뺐기 때문이다.

정리하면

노란색 구간에서 빨강과 주황구간이 사라지고 초록 구간이 더해지면 (2,2)~(3,4)까지의 구간만 남게된다.

코드

```
#include <iostream>
#include <vector>

using namespace std;

int main(){
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);

    int N, M;    cin >> N >> M;
    vector<vector<int>> arr(N+1, vector<int> (N+1,0));
    vector<vector<int>> sum(N+1, vector<int> (N+1,0));
    for(int i=1; i<N+1; i++){
        for(int j=1; j<N+1; j++){
            cin >> arr[i][j];
        }
    }

    //1행,1열 구간합
    for(int i=1; i<N+1; i++){
        sum[1][i] = sum[1][i-1] + arr[1][i];
        sum[i][1] = sum[i-1][1] + arr[i][1];
    }
    //2,2부터의 구간합
    for(int i=2; i<N+1; i++){
        for(int j=2; j<N+1; j++){
            sum[i][j] = sum[i][j-1] + sum[i-1][j] + arr[i][j] - sum[i-1][j-1];
        }
    }

    //구간 입력을 받자
    for(int testcase = 0; testcase < M; testcase++){
        int x_1,y_1,x_2,y_2;    cin >> y_1 >> x_1 >> y_2 >> x_2;
        cout << sum[y_2][x_2] - sum[y_1-1][x_2] - sum[y_2][x_1-1] + sum[y_1-1][x_1-1] << '\n';
    }
}
```