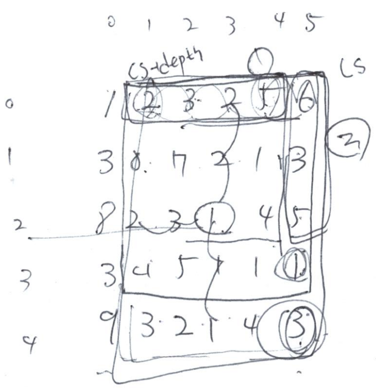
Baekjoon 17406 배열 돌리기 4

날짜	@2021년 8월 11일
링크	
열	
태그	Baekjoon Gold4 matrix

@2021년 8월 11일 for 풀었는데 배열 인덱스 맞추다가 간만에 머리에서 쥐났다.... 생각이 없다. 왜냐하면 생각이 없기 때무니이나의나ㅡ이나ㅡㅇ/.....

Baek 17406 골드 4



3 4 2

1 , 2 → 0,1 (cs rs)

5 , 6 → 4,5 (ce re)

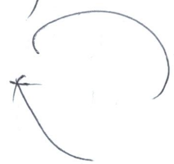
4 , 2 , 1

3 , 1 2 , 0

5 , 3 4 , 2

r c s

O(r-s, c-s)



1칸

O(r+s, c+s)

1 8 2 3 2 5
3 2 3 7 2 6
8 4 5 1 1 3
3 3 1 1 4 5
4 2 1 4 3 1

3 8 4
9 3 5
2 1 1

3, 4, 2

1, 2

5, 6

```
import java.io.BufferedReader;
import java.io.FileInputStream;
```

```
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.LinkedList;
import java.util.StringTokenizer;

//public class Main {
public class Baek17406_1 {
  static int N, M, K;
  static int[][] num;
  static int[][] map;
  static int[][] com;
  static int[][] command;
  static boolean[] chk;
  static ArrayList<LinkedList> a = new ArrayList<>();;
  static ArrayList b = new ArrayList<>();
  static int min = Integer.MAX_VALUE;

  public static void main(String[] args) throws IOException {
    System.setIn(new FileInputStream("C:/CodingStudy/Baekjoon/Gold4/17406_input.tx
t"));
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    StringTokenizer st = new StringTokenizer(br.readLine());
    N = Integer.parseInt(st.nextToken());
    M = Integer.parseInt(st.nextToken());
    K = Integer.parseInt(st.nextToken());
    com = new int[K][3];
    command = new int[K][3];
    num = new int[N][M];
    map = new int[N][M];
    chk = new boolean[K];

    for (int n = 0; n < N; n++) {
      st = new StringTokenizer(br.readLine());
      for (int m = 0; m < M; m++) {
        map[n][m] =num[n][m] = Integer.parseInt(st.nextToken());
      }
    } // end 초기 num, ans 입력

    for (int idx = 0; idx < K; idx++) {
      st = new StringTokenizer(br.readLine());
      for (int row = 0; row < 3; row++) {
        command[idx][row] = Integer.parseInt(st.nextToken());
      }
    } // end command 입력

    makeCommand(0);

    System.out.println(min);

//
    br.close();

  }// end main

  public static void makeCommand(int idx) {
    if (idx == K) {
```

```
      for (int n = 0; n < N; n++) {
        for (int m = 0; m < M; m++) {
          num[n][m] =map[n][m];
        }
      } // end 초기 num, ans 입력

      for (int x = 0; x < K; x++) {
        for (int i = 0; i < com[x][2]; i++) {
          a.add(new LinkedList<Integer>());
        } // end 각 테두리를 관리할 링크드 리스트 생성
        readLine(0, x); // 테두리 별로 링크드 리스트에 넣기
      }

      for (int q = 0; q < N; q++) {
        int sum = 0;
        for (int row = 0; row < M; row++) {
          sum += num[q][row];
        }
        if (sum < min)
          min = sum;
      }

      return;
    }

    for (int i = 0; i < K; i++) {
      if (chk[i])
        continue;
      com[idx] = command[i];
      chk[i] = true;
      makeCommand(idx + 1);
      chk[i] = false;

    }

  }

  public static void readLine(int depth, int idx) {// 테두리 별로 링크드 리스트에 넣기

    int rs = com[idx][1] - com[idx][2] - 1;
    int cs = com[idx][0] - com[idx][2] - 1;
    int re = com[idx][1] + com[idx][2] - 1;
    int ce = com[idx][0] + com[idx][2] - 1;

    for (int row = rs + depth; row <= re - 1 - depth; row++) {
      a.get(depth).add(num[cs + depth][row]);
    }
    for (int col = cs + depth; col <= ce - 1 - depth; col++) {
      a.get(depth).add(num[col][re - depth]);
    }
    for (int row = re - depth; row > rs + depth; row--) {
      a.get(depth).add(num[ce - depth][row]);
    }
    for (int col = ce - depth; col > cs + depth; col--) {
      a.get(depth).add(num[col][rs + depth]);
    }

    a.get(depth).offerFirst(a.get(depth).pollLast()); // 1번 회전
```

```
    for (int row = rs + depth; row <= re - 1 - depth; row++) {
      num[cs + depth][row] = (int) a.get(depth).pop();
    }
    for (int col = cs + depth; col <= ce - 1 - depth; col++) {
      num[col][re - depth] = (int) a.get(depth).pop();
    }
    for (int row = re - depth; row > rs + depth; row--) {
      num[ce - depth][row] = (int) a.get(depth).pop();
    }
    for (int col = ce - depth; col > cs + depth; col--) {
      num[col][rs + depth] = (int) a.get(depth).pop();
    }

//    System.out.println("depth : " + depth);
//    for (int i = 0; i < N; i++) {
//      for (int j = 0; j < M; j++) {
//        System.out.print(num[i][j] + " ");
//      }
//      System.out.println();
//    }

    if (depth == com[idx][2] - 1)
      return;
    readLine(depth + 1, idx);
  }
}
```