

1. 문제 요약

1. 빙고 게임을 한다.
2. 빙고가 3개 이상 되는 순간에 몇 번째 수를 부른 후 인지 출력

2. 문제 입/출력

입력

1. 첫째 줄부터 다섯째 줄까지 빙고판에 쓰여진 수가 가장 위 가로줄부터 차례대로 한 줄에 다섯 개씩 빈 칸을 사이에 두고 주어진다.
2. 여섯째 줄부터 열째 줄까지 사회자가 부르는 수가 차례대로 한 줄에 다섯 개씩 빈 칸을 사이에 두고 주어진다.
3. 빙고판에 쓰여진 수와 사회자가 부르는 수는 각각 1부터 25까지의 수가 한 번씩 사용된다.

출력

1. 첫째 줄에 사회자가 몇 번째 수를 부른 후 철수가 "빙고"를 외치게 되는지 출력한다.

3. 접근 아이디어

1. 각 행과 열, 대각선에 대한 DAT(Direct Access Table) 사용
2. 번호가 주어질 때마다 해당 배열에 count up 최대 500,000번 입력

4. 구현 코드

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.*;

public class Main {

    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new
        InputStreamReader(System.in));

        int[][] map = new int[26][2];
        StringTokenizer st;
        for(int y = 0; y < 5; y++) {
            st = new StringTokenizer(br.readLine());
            for(int x = 0; x < 5; x++) {
                int num = Integer.parseInt(st.nextToken());
                map[num][0] = x;
                map[num][1] = y;
            } // 자료의 저장은 x, y로
        }

        // DAT(Direct Access Table)
        int[] y = new int[5];
        int[] x = new int[5];
```

```
int[] diag = new int[2];

int result = 0, bingo = 0;
for(int i = 0; i < 5; i++) {
    st = new StringTokenizer(br.readLine());
    for(int j = 0; j < 5; j++) {
        int[] pos = map[Integer.parseInt(st.nextToken())];
        result++;

        x[pos[0]]++;
        if(x[pos[0]] == 5) bingo++;

        y[pos[1]]++;
        if(y[pos[1]] == 5) bingo++;

        if(pos[0] == pos[1]) {
            diag[0]++;
            if(diag[0] == 5) bingo++;
        }

        if(pos[0] + pos[1] == 4) {
            diag[1]++;
            if(diag[1] == 5) bingo++;
        }

        if(bingo >= 3) {
            System.out.println(result);
            return;
        }
    }
}
}
```