



# BOJ 14500 : 테트로미오

[통계](#) [수정](#) [삭제](#)

beberiche · 방금 전

0

## algorithms



1. BOJ 1593 : 문자해독
2. BOJ 11051 : 이항계수2
3. BOJ 2482 : 색상환
4. Programmers : 굴 고르기
5. BOJ 7569 : 토마토
6. **BOJ 14500 : 테트로미오**

▲ 숨기기

6/6



## 문제

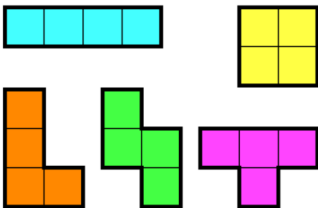
시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	512 MB	66517	24869	16171	35.509%

문제

폴리오미노란 크기가 1×1인 정사각형을 여러 개 이어서 붙인 도형이며, 다음과 같은 조건을 만족해야 한다.

- 정사각형은 서로 겹치면 안 된다.
- 도형은 모두 연결되어 있어야 한다.
- 정사각형의 변끼리 연결되어 있어야 한다. 즉, 꼭짓점과 꼭짓점만 맞닿아 있으면 안 된다.

정사각형 4개를 이어 붙인 폴리오미노는 테트로미노라고 하며, 다음과 같은 5가지가 있다.



아름이는 크기가 N×M인 종이 위에 테트로미노 하나를 놓으려고 한다. 종이는 1×1 크기의 칸으로 나누어져 있으며, 각각의 칸에는 정수가 하나 쓰여 있다.

테트로미노 하나를 적절히 놓아서 테트로미노가 놓인 칸에 쓰여 있는 수들의 합을 최대로 하는 프로그램을 작성하시오.

테트로미노는 반드시 한 정사각형이 정확히 하나의 칸을 포함하도록 놓아야 하며, 회전이나 대칭을 시켜도 된다.

입력

첫째 줄에 종이의 세로 크기 N과 가로 크기 M이 주어진다. (4 ≤ N, M ≤ 500)

둘째 줄부터 N개의 줄에 종이에 쓰여 있는 수가 주어진다. i번째 줄의 j번째 수는 위에서부터 i번째 칸, 왼쪽에서부터 j번째 칸에 쓰여 있는 수이다. 입력으로 주어지는 수는 1,000을 넘지 않는 자연수이다.

출력

첫째 줄에 테트로미노가 놓인 칸에 쓰인 수들의 합의 최댓값을 출력한다.

예제 입력 1 복사

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

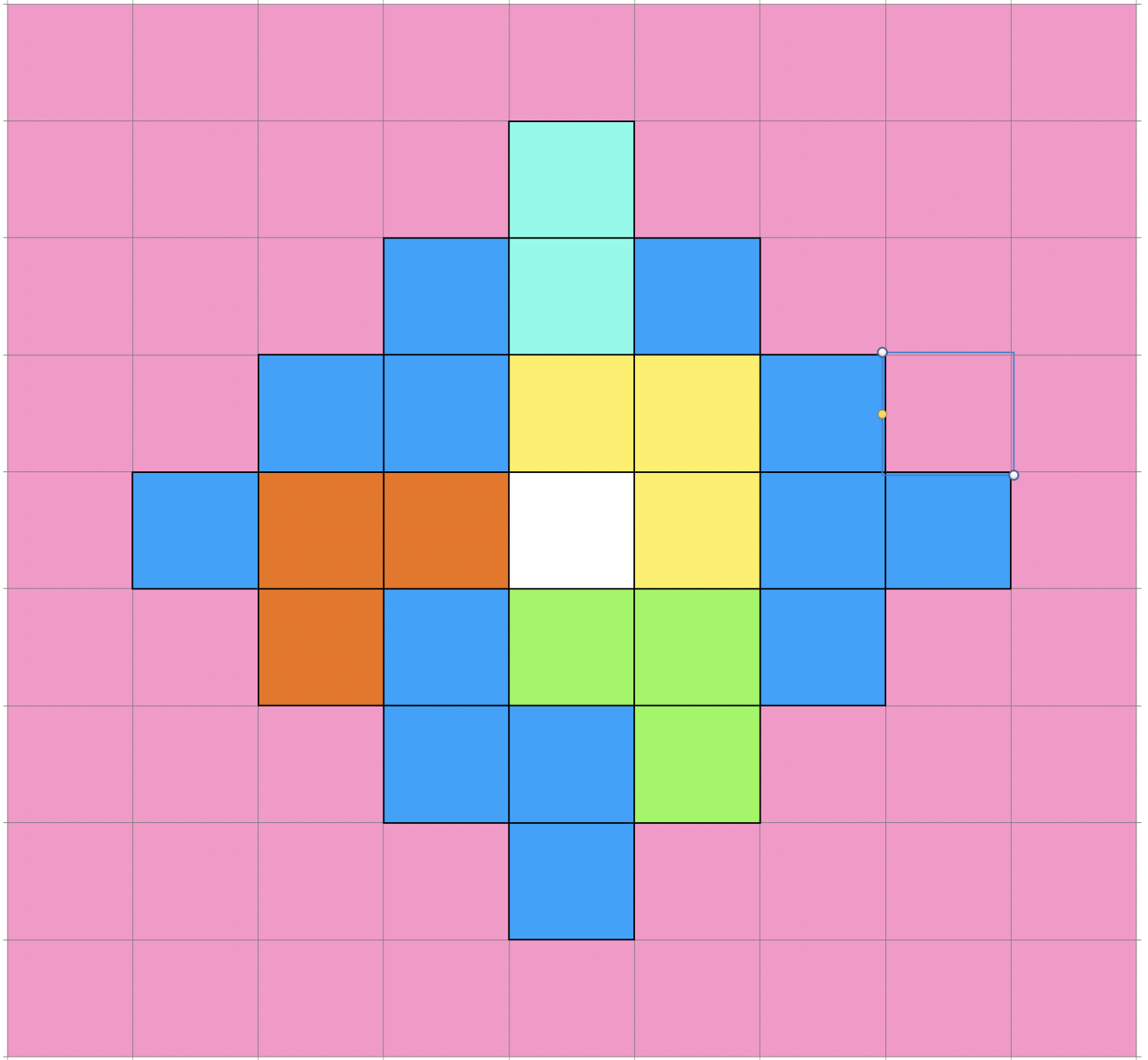
예제 출력 1 복사

```
19
```

# 풀이과정

뭔가 풀면서 그래도 배열 인덱스 맞추는 건 익숙해졌구나 했던 문제. 단순 배열 이동을 통한 브루트포스로 푸는 방법과, 다른 블로그를 보니 DFS 를 사용하는 방법이 있더라.

원리는 이러하다  $\perp$  모양을 제외한 모든 블록은 DFS(4) 에 모두 포함된다 DFS(4) 의 방향을 모두 경우가 테트로미노 블록 형태에 다 포함되기 때문에, 그냥 해당  $r, c$  를 지점으로 탐색하면 된다.



이런 걸 바로바로 생각할 수 있으려면 더 열심히 알고리즘을 풀어야 겠다.

참조 <https://velog.io/@skypepodium/%EB%B0%B1%EC%A4%80-14500-%ED%85%8C%ED%8A%B8%EB%A1%9C%EB%AF%B8%EB%85%B8>

여튼 나의 경우는 모두 나올 수 있는 19가지를 만들어서 그냥 배열을 돌렸다.

## 정답

```
import java.util.Scanner;

public class Main {
    // 정사각형
    static int[][] tr1 = {{0,1},{1,0},{1,1}};
    // 막대기
    static int[][] tr2Row = {{0,1},{0,2},{0,3}};
    static int[][] tr2Col = {{1,0},{2,0},{3,0}};
    // 번개모양?
    static int[][] tr3Row = {{0,1},{-1,1},{-1,2}};
    static int[][] tr3Col = {{1,0},{1,1},{2,1}};
    static int[][] tr3RowRev = {{0,1},{1,1},{1,2}};
    static int[][] tr3ColRev = {{1,0},{1,-1},{2,-1}};
    // 기억
    static int[][] tr4Ang0 = {{1,0},{2,0},{2,1}};
    static int[][] tr4Ang90 = {{1,0},{0,1},{0,2}};
    static int[][] tr4Ang180 = {{0,1},{1,1},{2,1}};
    static int[][] tr4Ang270 = {{1,0},{1,-1},{1,-2}};
    static int[][] tr4Ang0Rev = {{1,0},{2,0},{2,-1}};
    static int[][] tr4Ang90Rev = {{0,1},{0,2},{1,2}};
    static int[][] tr4Ang180Rev = {{0,1},{1,0},{2,0}};
    static int[][] tr4Ang270Rev = {{1,0},{1,1},{1,2}};
    // 성?
    static int[][] tr5Ang0 = {{0,1},{0,2},{1,1}};
    static int[][] tr5Ang90 = {{1,0},{2,0},{1,-1}};
    static int[][] tr5Ang180 = {{1,-1},{1,0},{1,1}};
    static int[][] tr5Ang270 = {{1,0},{2,0},{1,1}};

    static int[][] map;
    static int N,M;

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        N = sc.nextInt();
        M = sc.nextInt();

        map = new int[N][M];
        for(int i=0; i<N; i++) {
            for(int j=0; j<M; j++) {
                map[i][j] = sc.nextInt();
            }
        }

        System.out.println(findMaxSum());
    }
}
```

```

    }
    private static int findMaxSum() {
        int ans = 0;
        for(int r=0; r<N; r++) {
            for(int c=0; c<M; c++) {
                // 해당 r, c 기준으로 테트로미노 블록을 놓았을 때 가질 수 있는 합
                ans = Math.max(ans, currTetrominoSum(tr1, r, c));
                ans = Math.max(ans, currTetrominoSum(tr2Row, r, c));
                ans = Math.max(ans, currTetrominoSum(tr2Col, r, c));
                ans = Math.max(ans, currTetrominoSum(tr3Row, r, c));
                ans = Math.max(ans, currTetrominoSum(tr3Col, r, c));
                ans = Math.max(ans, currTetrominoSum(tr3RowRev, r, c));
                ans = Math.max(ans, currTetrominoSum(tr3ColRev, r, c));
                ans = Math.max(ans, currTetrominoSum(tr4Ang0, r, c));
                ans = Math.max(ans, currTetrominoSum(tr4Ang90, r, c));
                ans = Math.max(ans, currTetrominoSum(tr4Ang180, r, c));
                ans = Math.max(ans, currTetrominoSum(tr4Ang270, r, c));
                ans = Math.max(ans, currTetrominoSum(tr4Ang0Rev, r, c));
                ans = Math.max(ans, currTetrominoSum(tr4Ang90Rev, r, c));
                ans = Math.max(ans, currTetrominoSum(tr4Ang180Rev, r, c));
                ans = Math.max(ans, currTetrominoSum(tr4Ang270Rev, r, c));
                ans = Math.max(ans, currTetrominoSum(tr5Ang0, r, c));
                ans = Math.max(ans, currTetrominoSum(tr5Ang90, r, c));
                ans = Math.max(ans, currTetrominoSum(tr5Ang180, r, c));
                ans = Math.max(ans, currTetrominoSum(tr5Ang270, r, c));
            }
        }
        return ans;
    }

    private static int currTetrominoSum(int[][] tetromino, int r, int c) {
        int sum = map[r][c];
        for(int k=0; k<3; k++) {
            int nr = r+tetromino[k][0];
            int nc = c+tetromino[k][1];

            if(nr<0 || nc<0 || nr>=N || nc>=M) return 0;

            sum += map[nr][nc];
        }
        return sum;
    }
}

```



**벨**

새로운 것에 관심이 많고, 프로젝트 설계 및 최적화를 좋아합니다.





이전 포스트

**BOJ 7569 : 토마토**

0개의 댓글

댓글을 작성하세요

댓글 작성