



# #16509. 장군

<https://www.acmicpc.net/problem/16509>

25.05.26



# Problem

시간: 1시간 10분

16509번

제출

맞힌 사람

숫코딩

재채점 결과

채점 현황

내 제출

난이도 기여

질문 게시판

장군

성공



시간 제한

메모리 제한

제출

정답

맞힌 사람

정답 비율

1 초

512 MB

2019

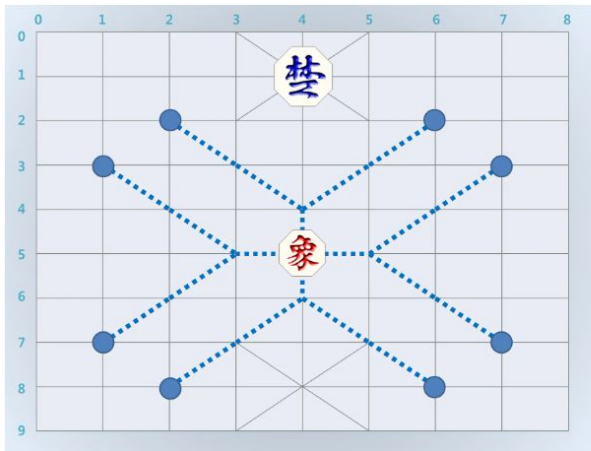
1082

910

54.393%

## 문제

오랜만에 휴가를 나온 호근이는 문득 동아리방에 있는 장기가 하고 싶어졌다. 하지만 장기를 오랫동안 하지 않은 탓인지 예전에는 잘 쓰던 상을 제대로 쓰는 것이 너무 힘들었다. 호근이를 위해 상을 어떻게 써야 할지 도와주자.

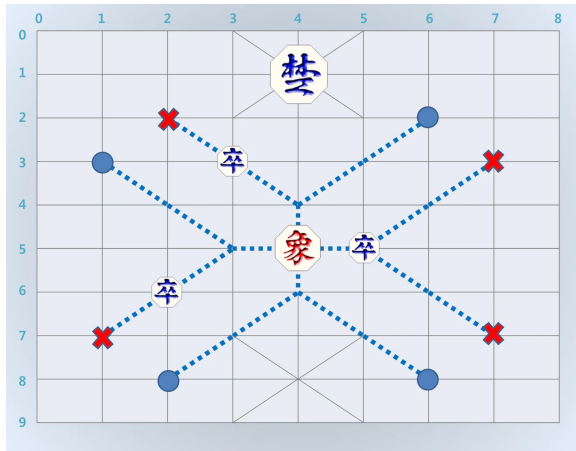




# Problem

## 문제 요구 사항 정리

1. 10×9 크기의 장기판에서 상으로 왕을 잡아야 한다.
2. 상은 위 그림과 같이 8가지 방법으로 움직일 수 있다. 상, 하, 좌, 우로 한 칸 이동 후 같은 방향 쪽 대각선으로 두 칸 이동한다.
3. 만약 상이 이동하는 경로에 다른 기물이(왕 포함) 있다면 상은 그쪽으로 이동할 수 없다. 또한, 상이 장기판을 벗어날 수도 없다.
4. 상이 왕에게 도달할 수 있는 최소 이동 횟수를 출력한다. 만약 도달할 수 없다면 -1을 출력한다.





# Hint

## Step 1.

4. 상이 왕에게 도달할 수 있는 **최소 이동 횟수**를 출력하라

- 상은 이동할 수 있는 경로가 여러 개 있으며, 최단 이동 횟수를 찾는 문제이므로 **최단 경로 알고리즘**인 BFS(Breadth-First Search) 를 사용.
- 시간 복잡도
  - a. **BFS 탐색**: 각 위치에서 최대 8가지 방향으로 이동.
  - b. **장기판의 전체 크기**:  $10 \times 9 = 90$ .
  - c. **총 연산**: 각 위치당 최대 8번의 이동을 시도하므로,  $O(8 * 90) = O(720)$ 입니다.
  - d. **실제 수행 시간**: 매우 작으며, 1초안에 해결 가능.



# Hint

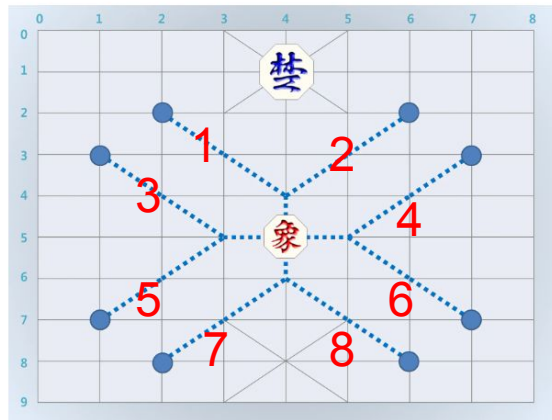
## Step 2.

2. 상은 8가지 방향으로 움직일 수 있다.

- 장기판 위에 상을 이동시키는 시뮬레이션을 진행할 때 방향 벡터 필요. 각각 1 ~ 8번 순서대로 정의.

○  $dx = [-3, -3, -2, -2, 2, 2, 3, 3]$

○  $dy = [-2, 2, -3, 3, -3, 3, -2, 2]$





# Hint

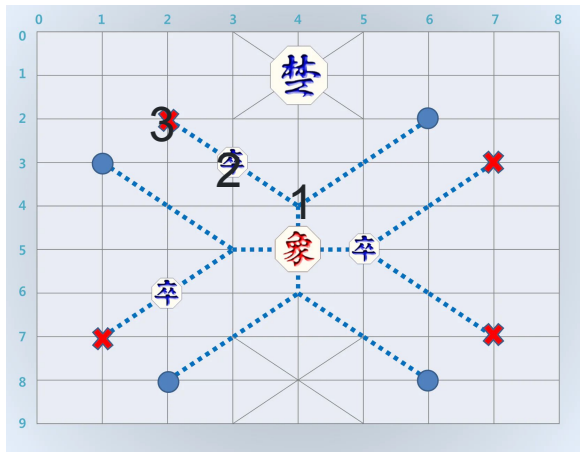
## Step 3.

3. 상이 이동하는 경로에 다른 기물이(왕 포함) 존재하면 이동 불가

- 장기판 위에 상을 이동시킬 때, 총 3번의 장기판을 거쳐 이동. tx, ty는 상의 8가지 이동 방향 및 그에 따른 경유 좌표를 정의한다. tx는 각각 [경유 1의 dx, 경유 2의 dx]

```
tx = [[-1, -2], [-1, -2], [0, -1], [0, -1], [0, 1], [0, 1], [1, 2], [1, 2]]
```

```
ty = [[0, -1], [0, 1], [-1, -2], [1, 2], [-1, -2], [1, 2], [0, -1], [0, 1]]
```



# 🌟🌟Solution

```
from collections import deque
input = import ('sys').stdin.readline
arr = [[0] * 9 for i in range(10)]
visit = [[0] * 9 for i in range(10)]
dx = [-3, -3, -2, -2, 2, 2, 3, 3]
dy = [-2, 2, -3, 3, -3, 3, -2, 2]
r, c = map(int, input().split())
a, b = map(int, input().split())
arr[a][b] = 1
```

```
def move(i, j, tp):
```

```
    tx = [
```

```
        [-1, -2],
        [-1, -2],
        [0, -1],
        [0, -1],
        [0, 1],
        [0, 1],
        [1, 2],
        [1, 2]
```

```
    ]
```

```
    ty = [
```

```
        [0, -1],
        [0, 1],
        [-1, -2],
        [1, 2],
        [-1, -2],
        [1, 2],
        [0, -1],
        [0, 1]
```

```
# 상의 이동경로에 왕이 존재하면 갈 수 없음: 0 반환
```

```
    for k in range(2):
```

```
        x, y = i + tx[tp][k], j + ty[tp][k]
```

```
        if (x == a and y == b):
```

```
            return 0
```

```
    return 1
```

```
def bfs(i, j):
```

```
    q = deque()
```

```
    visit[i][j] = 1
```

```
    q.append((i, j, 0))
```

```
    while q:
```

```
        i, j, c = q.popleft()
```

```
        for k in range(8):
```

```
            x, y = i + dx[k], j + dy[k]
```

```
            if not(0 <= x < 10 and 0 <= y < 9): continue
```

```
            if not move(i, j, k): continue
```

```
            if visit[x][y]: continue
```

```
            if x == a and y == b: return c + 1
```

```
            visit[x][y] = 1
```

```
            q.append((x, y, c + 1))
```

```
    return -1
```

```
print(bfs(r, c))
```



# Assignment

## 백준 #11559. Puyo Puyo (골드 4)

복잡한 시뮬레이션 문제 & 그래프 탐색 활용 문제