**문제 설명**

문제 설명

플레이어 A와 플레이어 B가 서로 게임을 합니다. 당신은 이 게임이 끝날 때까지 양 플레이어가 캐릭터를 몇 번 움직이게 될지 예측하려고 합니다.

각 플레이어는 자신의 캐릭터 하나를 보드 위에 올려놓고 게임을 시작합니다. 게임 보드는 1x1 크기 정사각 격자로 이루어져 있으며, 보드 안에는 발판이 있는 부분과 없는 부분이 있습니다. 발판이 있는 곳에만 캐릭터가 서있을 수 있으며, 처음 캐릭터를 올려놓는 곳은 항상 발판이 있는 곳입니다. 캐릭터는 발판이 있는 곳으로만 이동할 수 있으며, 보드 밖으로 이동할 수 없습니다. 밟고 있던 발판은 그 위에 있던 캐릭터가 다른 곳으로 이동하여 다른 발판을 밞음과 동시에 사라집니다. 양 플레이어는 번갈아가며 자기 차례에 자신의 캐릭터를 상하좌우로 인접한 4개의 칸 중에서 발판이 있는 칸으로 옮겨야 합니다.

다음과 같은 2가지 상황에서 패자와 승자가 정해지며, 게임이 종료됩니다.

* 움직일 차례인데 캐릭터의 상하좌우 주변 4칸이 모두 발판이 없거나 보드 밖이라서 이동할 수 없는 경우, 해당 차례 플레이어는 패배합니다.
* 두 캐릭터가 같은 발판 위에 있을 때, 상대 플레이어의 캐릭터가 다른 발판으로 이동하여 자신의 캐릭터가 서있던 발판이 사라지게 되면 패배합니다.

게임은 항상 플레이어 A가 먼저 시작합니다. 양 플레이어는 최적의 플레이를 합니다. 즉, 이길 수 있는 플레이어는 최대한 빨리 승리하도록 플레이하고, 질 수밖에 없는 플레이어는 최대한 오래 버티도록 플레이합니다. '이길 수 있는 플레이어'는 실수만 하지 않는다면 항상 이기는 플레이어를 의미하며, '질 수밖에 없는 플레이어'는 최선을 다해도 상대가 실수하지 않으면 항상 질 수밖에 없는 플레이어를 의미합니다. 최대한 오래 버틴다는 것은 양 플레이어가 캐릭터를 움직이는 횟수를 최대화한다는 것을 의미합니다.

아래 그림은 초기 보드의 상태와 각 플레이어의 위치를 나타내는 예시입니다.

위와 같은 경우, 플레이어 A는 실수만 하지 않는다면 항상 이길 수 있습니다. 따라서 플레이어 A는 이길 수 있는 플레이어이며, B는 질 수밖에 없는 플레이어입니다. 다음은 A와 B가 최적의 플레이를 하는 과정을 나타냅니다.

1. 플레이어 A가 초기 위치 (1, 0)에서 (1, 1)로 이동합니다. **플레이어 A가 (0, 0)이나 (2, 0)으로 이동할 경우 승리를 보장할 수 없습니다. 따라서 무조건 이길 방법이 있는 (1, 1)로 이동합니다.**
2. 플레이어 B는 (1, 1)로 이동할 경우, 바로 다음 차례에 A가 위 또는 아래 방향으로 이동하면 발판이 없어져 패배하게 됩니다. **질 수밖에 없는 플레이어는 최대한 오래 버티도록 플레이하기 때문에 (1, 1)로 이동하지 않습니다.** (1, 2)에서 위쪽 칸인 (0, 2)로 이동합니다.
3. A가 (1, 1)에서 (0, 1)로 이동합니다.
4. B에게는 남은 선택지가 (0, 1)밖에 없습니다. 따라서 (0, 2)에서 (0, 1)로 이동합니다.
5. A가 (0, 1)에서 (0, 0)으로 이동합니다. 이동을 완료함과 동시에 B가 서있던 (0, 1)의 발판이 사라집니다. B가 패배합니다.
6. 만약 과정 2에서 B가 아래쪽 칸인 (2, 2)로 이동하더라도 A는 (2, 1)로 이동하면 됩니다. 이후 B가 (2, 1)로 이동, 다음 차례에 A가 (2, 0)으로 이동하면 B가 패배합니다.

위 예시에서 양 플레이어가 최적의 플레이를 했을 경우, 캐릭터의 이동 횟수 합은 5입니다. 최적의 플레이를 하는 방법은 여러 가지일 수 있으나, 이동한 횟수는 모두 5로 같습니다.

게임 보드의 초기 상태를 나타내는 2차원 정수 배열 board와 플레이어 A의 캐릭터 초기 위치를 나타내는 정수 배열 aloc, 플레이어 B의 캐릭터 초기 위치를 나타내는 정수 배열 bloc이 매개변수로 주어집니다. 양 플레이어가 최적의 플레이를 했을 때, 두 캐릭터가 움직인 횟수의 합을 return 하도록 solution 함수를 완성해주세요.

제한사항

* 1 ≤ board의 세로 길이 ≤ 5
* 1 ≤ board의 가로 길이 ≤ 5
* board의 원소는 0 또는 1입니다.
  + 0은 발판이 없음을, 1은 발판이 있음을 나타냅니다.
  + 게임 보드의 좌측 상단 좌표는 (0, 0), 우측 하단 좌표는 (board의 세로 길이 - 1, board의 가로 길이 - 1)입니다.
* aloc과 bloc은 각각 플레이어 A의 캐릭터와 플레이어 B의 캐릭터 초기 위치를 나타내는 좌표값이며 [r, c] 형태입니다.
  + r은 몇 번째 행인지를 나타냅니다.
  + 0 ≤ r < board의 세로 길이
  + c는 몇 번째 열인지를 나타냅니다.
  + 0 ≤ c < board의 가로 길이
  + 초기 보드의 aloc과 bloc 위치는 항상 발판이 있는 곳입니다.
  + aloc과 bloc이 같을 수 있습니다.
* 상대 플레이어의 캐릭터가 있는 칸으로 이동할 수 있습니다.

입출력 예

| **board** | **aloc** | **bloc** | **result** |
| --- | --- | --- | --- |
| [[1, 1, 1], [1, 1, 1], [1, 1, 1]] | [1, 0] | [1, 2] | 5 |
| [[1, 1, 1], [1, 0, 1], [1, 1, 1]] | [1, 0] | [1, 2] | 4 |
| [[1, 1, 1, 1, 1]] | [0, 0] | [0, 4] | 4 |
| [[1]] | [0, 0] | [0, 0] | 0 |

입출력 예 설명

**입출력 예 #1**

문제 예시와 같습니다.

**입출력 예 #2**

주어진 조건을 그림으로 나타내면 아래와 같습니다.

항상 이기는 플레이어는 B, 항상 지는 플레이어는 A입니다.

다음은 B가 이기는 방법 중 하나입니다.

1. A가 (1, 0)에서 (0, 0)으로 이동
2. B가 (1, 2)에서 (2, 2)로 이동
3. A가 (0, 0)에서 (0, 1)로 이동
4. B가 (2, 2)에서 (2, 1)로 이동
5. A가 (0, 1)에서 (0, 2)로 이동
6. B가 (2, 1)에서 (2, 0)으로 이동
7. A는 더 이상 이동할 수 없어 패배합니다.

위와 같이 플레이할 경우 이동 횟수 6번 만에 게임을 B의 승리로 끝낼 수 있습니다.

B가 다음과 같이 플레이할 경우 게임을 더 빨리 끝낼 수 있습니다. 이길 수 있는 플레이어는 최대한 빨리 게임을 끝내려 하기 때문에 위 방법 대신 아래 방법을 선택합니다.

1. A가 (1, 0)에서 (0, 0)으로 이동
2. B가 (1, 2)에서 (0, 2)로 이동
3. A가 (0, 0)에서 (0, 1)로 이동
4. B가 (0, 2)에서 (0, 1)로 이동
5. A는 더 이상 이동할 수 없어 패배합니다.

위와 같이 플레이할 경우 이동 횟수 4번 만에 게임을 B의 승리로 끝낼 수 있습니다. 따라서 4를 return 합니다.

**입출력 예 #3**

양 플레이어는 매 차례마다 한 가지 선택지밖에 고를 수 없습니다. 그 결과, (0, 2)에서 어디로도 이동할 수 없는 A가 패배합니다. 양 플레이어가 캐릭터를 움직인 횟수의 합은 4입니다.

**입출력 예 #4**

게임을 시작하는 플레이어 A가 처음부터 어디로도 이동할 수 없는 상태입니다. 따라서 A의 패배이며, 이동 횟수의 합은 0입니다.