**금 채굴하기**

**90XP**

**보통**

정답률 30%·제출 2,700회·예상 소요 시간 180분 이상

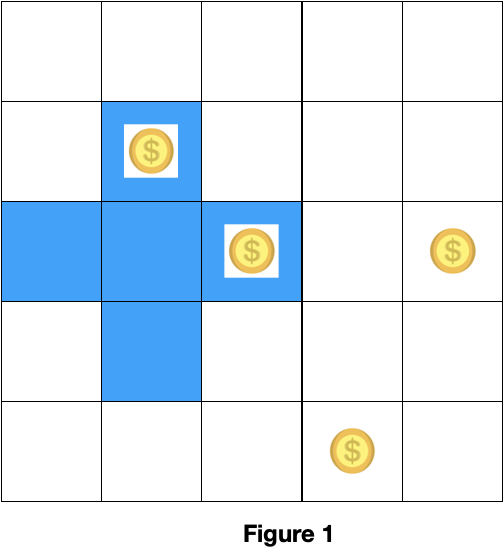
12아쉬워요

1 StarEmpty내 리스트에 추가

*n*×*n*크기의 이차원 영역에 파묻힌 금을 손해를 보지 않는 선에서 최대한 많이 채굴하려고 합니다. 채굴은 반드시 [그림 1, 2]과 같은 마름모 모양으로 단 한 번 할 수 있으며, 마름모 모양을 지키는 한 [그림 3]와 같이 이차원 영역을 벗어난 채굴도 가능하지만 이차원 영역 밖에 금은 존재하지 않습니다.

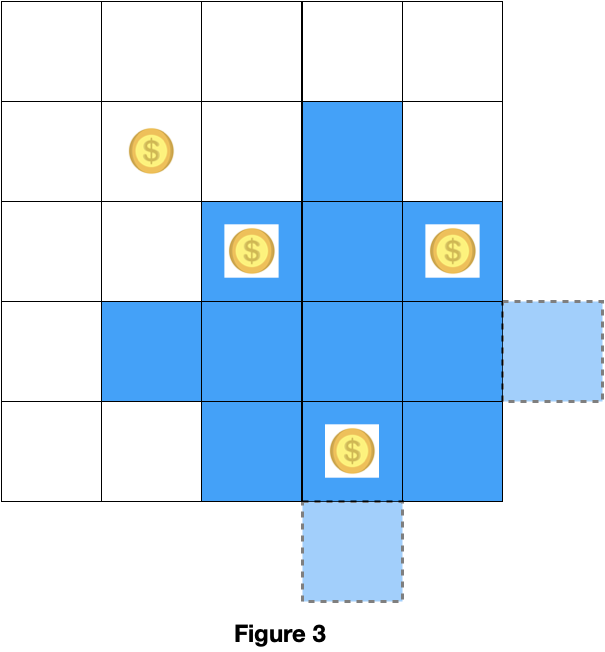
여기서 마름모 모양이란 특정 중심점을 기준으로 *K*번 이내로 상하좌우의 인접한 곳으로 이동하는 걸 반복했을 때 갈 수 있는 모든 영역이 색칠되어 있는 모양을 의미합니다. [그림 1]은 *K*가 1일때의 마름모 모양이고, [그림 2]는 *K*가 2일때 마름모 모양입니다. *K*가 0인 경우는 지점 한 곳에서만 채굴하는 것을 의미하며 이 역시 올바른 마름모 모양이라 할 수 있습니다. 올바르지 않은 마름모 모양을 이용해서는 채굴이 불가능합니다.

이 때 채굴에 드는 비용은 마름모 안의 격자 갯수만큼 들어가며, 이는 *K*∗*K*+(*K*+1)∗(*K*+1)로 계산될 수 있습니다. 금 한 개의 가격이 *m*일 때, 손해를 보지 않으면서 채굴할 수 있는 가장 많은 금의 개수를 출력하는 코드를 작성해보세요. 단 한 개의 격자 안에는 최대 한 개의 금만 존재합니다



광장이(가) 표시된 사진

자동 생성된 설명



**입력 형식**

첫 번째 줄에는 *n*과 *m*이 공백을 사이에 두고 주어지고,

두 번째 줄부터 (*n*+1)번째 줄까지는 각 행에 금이 있는 경우 1, 없는 경우 0으로 입력이 공백을 사이에 두고 주어집니다.

* 1≤*n*≤20
* 1≤*m*≤10

**출력 형식**

손해를 보지 않으면서 채굴할 수 있는 가장 많은 금의 개수를 출력해줍니다.

**입출력 예제**

**예제1**

입력:

5 5

0 0 0 0 0

0 1 0 0 0

0 0 1 0 1

0 0 0 0 0

0 0 0 1 0

출력:

3

**예제2**

입력:

3 2

0 1 0

1 0 1

0 0 0

출력:

3

**예제 설명**

예제 1에서는 [그림 1] 오른쪽 그림과 같이 (3,3) 위치에 *K*가 2인 마름모를 그렸을 때, 채굴에 드는 비용은 13이고 금 3개의 가격은 15 이므로 손해를 보지 않으면서 3개의 금을 얻을 수 있게 됩니다.

손해를 보지 않으면서 금 4개를 얻을 수 있는 방법은 존재하지 않습니다.

**제한**

시간 제한: 1000ms

메모리 제한: 80MB