

아시달리아에서 스키아파렐리까지

화성 탐사 팀 아레스 3의 멤버인 마크 와트니는 화성에 지나치게 충분히 오래 있었습니다. 이제 화성을 떠나기 위해 아레스 3의 탐사기지가 있는 아시달리아 평원에서 아레스 4의 MAV가 있는 스키아파렐리 분화구까지 가야 합니다.

이를 위해 와트니는 로버에 생명유지장비를 달고 배터리를 증설하고 태양전지판을 실었습니다. 그리고 최대한 안전하게 가기 위해 모래폭풍 예보와 지형 지도를 받았습니다.

이제 와트니는 최적의 동선을 판단해야 합니다. 가능한 빨리 가면 좋겠지만, 안전을 위해서는 전력을 최대한 유지하면서 위험한 지형을 피하는 것이 중요합니다. 이를 위해 와트니는 모래폭풍 예보를 해당 위치에서의 발전량으로 변환하고, 지형 지도를 전력소모량 지도로 변환했습니다.

이 지도를 가지고, 스키아파렐리에 도착했을 때 가능한 최대 전력량을 계산하세요.

-조건

- 배터리의 용량 Batt, 지도 크기 Col Row, 전력충전량 지도, 전력소모량 지도가 입력됩니다. 지도의 최대 크기는 16 x 12입니다.
- 로버의 전력량은 배터리의 용량을 넘을 수 없습니다. 예를 들어 배터리 용량이 10일 때, 10에서 출발해서 5를 쓰고 6을 충전하면 전력량은 11이 아닌 10이 됩니다.
- 로버는 출발할 때 완충하고 출발합니다.
- 전력충전량, 전력소모량은 모두 0보다 크거나 같습니다.
- 출발지점과 도착지점의 전력충전량은 0입니다.
- 출발지점의 전력소모량은 0입니다. 도착지점의 전력소모량은 0이 아닙니다.
- 출발점은 입력된 행렬의 왼쪽 위 끝점, 도착점은 오른쪽 아래 끝점입니다.
- (중요) 로버는 \rightarrow , \downarrow , \searrow 세 방향으로 움직일 수 있습니다.
- (중요) \searrow 방향으로 움직이는 경우, 전력소모량이 1.4배 증가합니다. 예를 들어 1,1 위치의 전력소모량이 5인 경우 다음과 같습니다.
 - (0,1), (1,0)에서 이동한 경우 : 전력소모량 5
 - (0,0)에서 이동한 경우 : 전력소모량 $5 * 1.4 = 7$
- (중요) 로버는 이동하고 나서 충전합니다. 이동 직후에 전력 잔량이 음수가 되면 안 됩니다.
- 모든 문제의 정답은 정수로 나옵니다. 소수점이 나오지 않도록 값이 설정되어 있습니다.

-입력

```
10          // 배터리의 용량 Batt
5 4         // 지형 지도의 규모 Col, Row(가로, 세로 순서)
0 6 6 6 6   // 위치별 전력충전량 지도 (Col개씩 Row줄)
6 6 3 3 6
6 3 3 3 6
3 1 1 3 0
0 5 5 400 5 // 지형에 따른 전력소모량 지도 (Col개씩 Row줄)
5 5 5 5 5   // 갈 수 없는 지점은 배터리 용량보다 큰 숫자로 표현
5 400 5 5 5
5 5 400 5 5
```

-출력

3

-해설과 힌트

각 지점에서는 다음 표와 같은 전력 잔량을 보유하게 됩니다. X는 도달할 수 없는 곳입니다.

각 지점에서의 전력충전량

0	6	6	6	6
6	6	3	3	6
6	3	3	3	6
3	1	1	3	0

각 지점에서의 전력소모량

0	5	5	400	5
5	5	5	5	5
5	400	5	5	5
5	5	400	5	5

각 지점에서의 전력 잔량

10	10	10	X	X
10	10	8	6	7
10	X	6	4	8
8	4	X	X	3

로버는 $\rightarrow \downarrow \searrow$ 방향으로만 진행할 수 있으므로, 전력 잔량에 대한 점화식을 구축하면 이 문제에 동적계획법을 적용할 수 있습니다. 정답은 전력 잔량 행렬의 가장 오른쪽 아래 값이 됩니다.

단, 점화식이 하나의 수식으로 나오지는 않습니다. 특히 이동 직후 충전하기 전에 로버의 전력이 음수가 될 수 없기 때문에 점화식에 조건 검사가 들어가야 합니다. 예를 들어 X 값을 갖는 (4,4) 위치의 값의 경우 (4,3)에서 출발하면 $4 \rightarrow -1(\text{이동 } -5) \rightarrow 2(\text{충전 } +3)$ 으로 최종적으로 양수 값이 나올 수 있으나, 이동하는 도중 전력 잔량이 0 미만이 되어서 생명유지장비가 꺼지기 때문에 이동할 수 없습니다.

(정답 소스는 다음 페이지에 있습니다. Java, C++)

Java

```
import java.util.Scanner;

public class AcidaliatoSchiaparelli {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int batt = scanner.nextInt();
        int col = scanner.nextInt();
        int row = scanner.nextInt();

        int[][] chargeArr = new int[row][col];
        int[][] consumeArr = new int[row][col];
        int[][] remainArr = new int[row][col];

        for(int r = 0; r < row; r++)
            for(int c = 0; c < col; c++)
                chargeArr[r][c] = scanner.nextInt();

        for(int r = 0; r < row; r++)
            for(int c = 0; c < col; c++)
                consumeArr[r][c] = scanner.nextInt();

        scanner.close();

        for(int r = 0; r < row; r++)
            for(int c = 0; c < col; c++) {
                if(r == 0 && c == 0) remainArr[r][c] = batt;
                else if(r == 0) {
                    int candidate0 = remainArr[r][c-1] - consumeArr[r][c];
                    if(candidate0 < 0) remainArr[r][c] = -99900;
                    else remainArr[r][c] = candidate0 + chargeArr[r][c];
                    if(remainArr[r][c] > batt) remainArr[r][c] = batt;
                }
                else if(c == 0) {
                    int candidate0 = remainArr[r-1][c] - consumeArr[r][c];
                    if(candidate0 < 0) remainArr[r][c] = -99900;
                    else remainArr[r][c] = candidate0 + chargeArr[r][c];
                    if(remainArr[r][c] > batt) remainArr[r][c] = batt;
                }
                else {
                    int candidate0 = remainArr[r-1][c] - consumeArr[r][c];
                    int candidate1 = remainArr[r][c-1] - consumeArr[r][c];
                    int candidate2 = remainArr[r-1][c-
1] - (int)(consumeArr[r][c]*1.4);
```

```
        int selected = candidate0;
        if(selected < candidate1) selected = candidate1;
        if(selected < candidate2) selected = candidate2;

        if(selected < 0) remainArr[r][c] = -99900;
        else remainArr[r][c] = selected + chargeArr[r][c];
        if(remainArr[r][c] > batt) remainArr[r][c] = batt;
    }
}

System.out.println(remainArr[row-1][col-1]);
}
```

C++

```
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
    int i, j, c, x, y, min[13][17], plus[13][17], ans[13][17];

    cin >> c;
    cin >> x >> y;

    for (i = 0; i < y; i++)
        for (j = 0; j < x; j++)
            cin >> plus[i][j];
    for (i = 0; i < y; i++)
        for (j = 0; j < x; j++)
            cin >> min[i][j];
    for (i = 0; i < y; i++)
        for (j = 0; j < x; j++)
            ans[i][j] = -1;

    ans[0][0] = c - min[0][0] + plus[0][0];
    if (ans[0][0] > c) ans[0][0] = c;

    for (i = 0; i < y; i++)
        for (j = 0; j < x; j++) {
            if (ans[i][j] != -1) {
                if (ans[i][j] - min[i][j + 1] >= 0) {//???
                    if (ans[i][j] - min[i][j + 1] + plus[i][j + 1] > ans[i][j + 1]) {
                        ans[i][j + 1] = ans[i][j] - min[i][j + 1] + plus[i][j + 1];
                        if (ans[i][j + 1] > c) ans[i][j + 1] = c;
                    }
                }

                if (ans[i][j] - min[i + 1][j] >= 0) {//3
                    if (ans[i][j] - min[i + 1][j] + plus[i + 1][j] > ans[i + 1][j]) {
                        ans[i + 1][j] = ans[i][j] - min[i + 1][j] + plus[i + 1][j];
                        if (ans[i + 1][j] > c) ans[i + 1][j] = c;
                    }
                }

                if (ans[i][j] - int(min[i + 1][j + 1] * 1.4) >= 0) {//반
```

```

        if (ans[i][j] - int(min[i + 1][j + 1] * 1.4) + plus[i + 1]
[j + 1] > ans[i + 1][j + 1]) {
            ans[i + 1][j + 1] = ans[i][j] - int(min[i + 1][j + 1]
* 1.4) + plus[i + 1][j + 1];
            if (ans[i + 1][j + 1] > c)ans[i + 1][j + 1] = c;
        }
    }
}

cout << ans[y-1][x-1];

return 0;
}

```