

TEST 14

1.

크기가 $N \times N$ 인 도시가 있습니다. 도시는 1×1 크기의 칸으로 나누어져 있습니다. 도시의 각 칸은 빈 칸, 치킨집, 집 중 하나입니다. 도시의 칸은 (r, c) 와 같은 형태로 나타내고, r 행 c 열 또는 위에서 서부터 r 번째 칸, 왼쪽에서부터 c 번째 칸을 의미합니다. r 과 c 는 1부터 시작합니다.

이 도시에 사는 사람들은 치킨을 매우 좋아합니다. 따라서, 사람들은 “치킨 거리” 라는 말을 주로 사용합니다. 치킨 거리는 집과 가장 가까운 치킨집 사이의 거리입니다. 즉, 치킨 거리는 집을 기준으로 정해지며, 각각의 집은 치킨 거리를 가지고 있습니다.

임의의 두 칸 (r_1, c_1) 과 (r_2, c_2) 사이의 거리는 $|r_1 - r_2| + |c_1 - c_2|$ 로 구합니다.

예를 들어, 아래와 같은 지도를 갖는 도시를 살펴봅시다.

0	2	0	1	0
1	0	1	0	0
0	0	0	0	0
0	0	0	1	1
0	0	0	1	2

0은 빈 칸, 1은 집, 2는 치킨집입니다.

$(2, 1)$ 에 있는 집과 $(1, 2)$ 에 있는 치킨집과의 거리는 $|2-1| + |1-2| = 2$, $(5, 5)$ 에 있는 치킨집과의 거리는 $|2-5| + |1-5| = 7$ 입니다. 따라서, $(2, 1)$ 에 있는 집의 치킨 거리는 2입니다.

$(5, 4)$ 에 있는 집과 $(1, 2)$ 에 있는 치킨집과의 거리는 $|5-1| + |4-2| = 6$, $(5, 5)$ 에 있는 치킨집과의 거리는 $|5-5| + |4-5| = 1$ 이다. 따라서, $(5, 4)$ 에 있는 집의 치킨 거리는 1이다.

이 도시에 있는 치킨집은 모두 같은 프랜차이즈입니다. 프랜차이즈 본사에서는 수익을 증가시키기 위해 일부 치킨집을 폐업시키려고 합니다. 오랜 연구 끝에 이 도시에서 가장 수익을 많이 낼 수 있는 치킨집의 개수는 최대 M 개라는 사실을 알아냈습니다.

도시에서 있는 치킨집 중에서 최대 M 개를 고르고, 나머지 치킨집은 모두 폐업시켜야 합니다. 어떻게 고르면, 도시의 치킨 거리가 가장 작게 될지 return하는 solution 함수를 작성해주세요.

TEST 14

제한사항

- ▶ N은 2이상 50 이하입니다. M은 1이상 13 이하입니다.
- ▶ 도시의 정보 city가 주어집니다. city는 0, 1, 2로 이루어져 있고, 0은 빈 칸, 1은 집, 2는 치킨집을 의미합니다. 집의 개수는 2N개를 넘지 않으며, 적어도 1개는 존재합니다. 치킨집의 개수는 M보다 크거나 같고, 13보다 작거나 같습니다.

N	M		return
5	3	[[0, 0, 1, 0, 0], [0, 0, 2, 0, 1], [0, 1, 2, 0, 0], [0, 0, 1, 0, 0], [0, 0, 0, 0, 2],]	5
5	3	[[1, 2, 0, 0, 0], [1, 2, 0, 0, 0], [1, 2, 0, 0, 0], [1, 2, 0, 0, 0], [1, 2, 0, 0, 0]]	11
5	3	[[0, 2, 0, 1, 0], [1, 0, 1, 0, 0], [0, 0, 0, 0, 0], [2, 0, 0, 1, 1], [2, 2, 0, 1, 2]]	10

TEST 14

2.

철수는 롤케이크를 두 조각으로 잘라서 동생과 한 조각씩 나눠 먹으려고 합니다. 이 롤케이크에는 여러가지 토핑들이 일렬로 올려져 있습니다. 철수와 동생은 롤케이크를 공평하게 나눠먹으려 하는데, 그들은 롤케이크의 크기보다 롤케이크 위에 올려진 토핑들의 종류에 더 관심이 많습니다. 그래서 잘린 조각들의 크기와 올려진 토핑의 개수에 상관없이 각 조각에 동일한 가짓수의 토핑이 올라가면 공평하게 롤케이크가 나누어진 것으로 생각합니다.

예를 들어, 롤케이크에 4가지 종류의 토핑이 올려져 있다고 합시다.

토핑들을 1, 2, 3, 4와 같이 번호로 표시했을 때, 케이크 위에 토핑들이 [1, 2, 1, 3, 1, 4, 1, 2] 순서로 올려져 있습니다.

만약 세 번째 토핑(1)과 네 번째 토핑(3) 사이를 자르면 롤케이크의 토핑은 [1, 2, 1], [3, 1, 4, 1, 2]로 나뉘게 됩니다. 철수가 [1, 2, 1]이 놓인 조각을, 동생이 [3, 1, 4, 1, 2]가 놓인 조각을 먹게 되면 철수는 두 가지 토핑(1, 2)을 맛볼 수 있지만, 동생은 네 가지 토핑(1, 2, 3, 4)을 맛볼 수 있으므로, 이는 공평하게 나누어진 것이 아닙니다.

만약 롤케이크의 네 번째 토핑(3)과 다섯 번째 토핑(1) 사이를 자르면 [1, 2, 1, 3], [1, 4, 1, 2]로 나뉘게 됩니다. 이 경우 철수는 세 가지 토핑(1, 2, 3)을, 동생도 세 가지 토핑(1, 2, 4)을 맛볼 수 있으므로, 이는 공평하게 나누어진 것입니다.

공평하게 롤케이크를 자르는 방법은 여러가지 일 수 있습니다. 위의 롤케이크를 [1, 2, 1, 3, 1], [4, 1, 2]으로 잘라도 공평하게 나눕니다. 어떤 경우에는 롤케이크를 공평하게 나누지 못할 수도 있습니다.

롤케이크에 올려진 토핑들의 번호를 저장한 정수 배열 `topping` 이 매개변수로 주어질 때, 롤케이크를 공평하게 자르는 방법의 수를 `return` 하도록 `solution` 함수를 완성해주세요.

TEST 14

제한사항

- ▶ $1 \leq \text{topping의 길이} \leq 1000000$
- ▶ $1 \leq \text{topping의 원소} \leq 10000$

입출력 예

topping	result
[1, 2, 1, 3, 1, 4, 1, 2]	2
[1, 2, 3, 1, 4]	0

입출력 예 설명

#1> 예시와 같습니다.

#2> 롤케이크를 공평하게 나눌 수 없습니다.

TEST 14

3. 초 단위로 기록된 주식 가격이 담긴 배열 prices가 매개변수로 주어질 때, 가격이 떨어지지 않은 기간은 몇 초인지를 return 하도록 solution 함수를 작성해주세요.

제한사항

- ▶ prices의 각 가격은 1 이상 10,000 이하인 자연수입니다.
- ▶ prices의 길이는 2 이상 100,000 이하입니다.

입출력 예

prices	return
1 2 3 2 3	4 3 1 1 0

입출력 예 설명

- > 1초 시점의 ₩1은 끝까지 가격이 떨어지지 않았습니다.
- > 2초 시점의 ₩2는 끝까지 가격이 떨어지지 않았습니다.
- > 3초 시점의 ₩3은 1초뒤에 가격이 떨어집니다. 따라서 1초간 가격이 떨어지지 않은 것으로 봅니다.
- > 4초 시점의 ₩2는 1초간 가격이 떨어지지 않았습니다.
- > 5초 시점의 ₩3은 0초간 가격이 떨어지지 않았습니다.