BOJ 5721.

사탕 줍기 대회

소프트웨어학과 201921017 이지우

Step1 – 접근

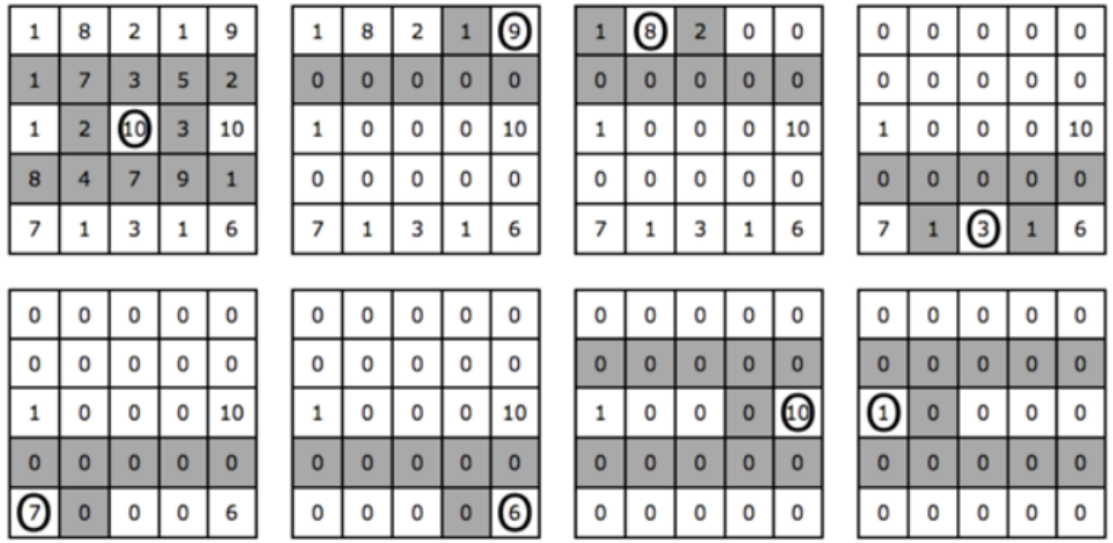
문제

상근이는 사탕에 중독된 아이이다. 상근이는 캔디 매거진의 열렬한 구독자이며, 올해 열리는 국제 사탕 줍기 대회에 한국 대표로 참가하게 되었다.

이 대회는 사탕을 포함하고 있는 박스가 M행 N열로 놓여져있는 곳에서 진행된다. (따라서 박스는 총 M × N개 있다) 각 박스에는 들어있는 사탕의 개수가 겉에 적혀져 있다.

대회의 참가자는 박스를 하나 고른다. 그 다음, 그 박스 안에 있는 사탕을 모두 가져가게 된다. 박스를 고르면, 고른 박스의 바로 위쪽 행과 바로 아래쪽 행에 있는 모든 박스, 그리고 고른 박스의 왼쪽과 오른쪽에 있는 박스에 들어있는 사탕이 모두 사라지게 된다. 참가자는 사탕이 들어있는 박스가 없을 때까지 박스를 고를 수 있다.

아래 그림을 살펴보자. 각 칸은 박스에 들어있는 사탕의 개수를 나타낸다. 각각의 단계에서 참가자가 고른 박스는 동그라미로 표시되어 있고, 회색으로 칠해진 박스는 참가자의 선택 때문에 사탕이 사라질 박스이다. 총 여덟 단계가 지나면 게임은 끝나게 되고, 상근이는 총 10+9+8+3+7+6+10+1 = 54개의 사탕을 가져가게 된다.

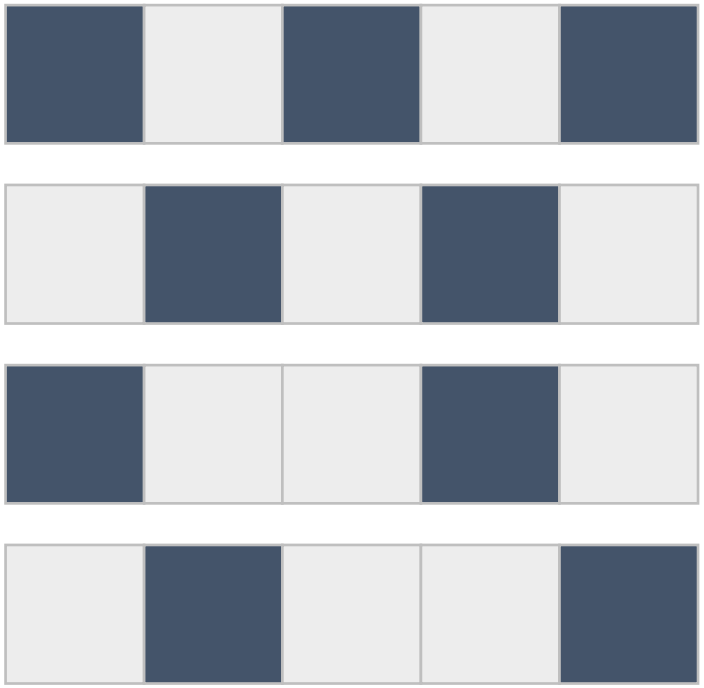


M과 N이 주어졌을 때, 상근이가 이 대회에서 가져갈 수 있는 사탕의 최대 개수를 구하는 프로그램을 작성하시오.

Step1 – 접근

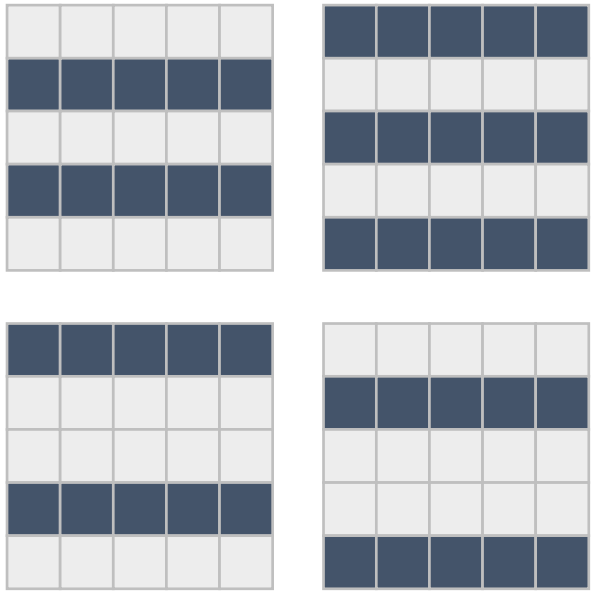
임의의 한 박스를 선택하면...

① 바로 양옆 박스들은 선택할 수 없게 됨
→ 사탕을 주울 수 있는 형태는...



첫 번째 박스가 선택됨/선택 안됨

② 바로 위, 아래 줄은 선택할 수 없게 됨
→ 사탕을 주울 수 있는 형태는...



첫 번째 줄이 선택됨/선택 안됨

Step1 – 접근: 어떤 **칸**을 선택해야 할까?

line	1	8	2	1	9

→ dp[] : 각 칸에 도달할 때까지 주울 수 있는 최대 사탕 수

→ 각 칸에 도달하는 방법
1. 두 번째 전 박스의 사탕을 줍고 자신의 사탕을 주움
2. 바로 직전 박스의 사탕을 줍고 자신의 사탕은 줍지 않음

→ 각 경우 얻을 수 있는 사탕 수를 비교하여 최대 값이 자신의 dp값이 되도록 함

dp	1	8	8	9	17

Step1 – 접근

lines

1	8	2	1	9
1	7	3	5	2
1	2	10	3	10
8	4	7	9	1
7	1	3	1	6

→

dp

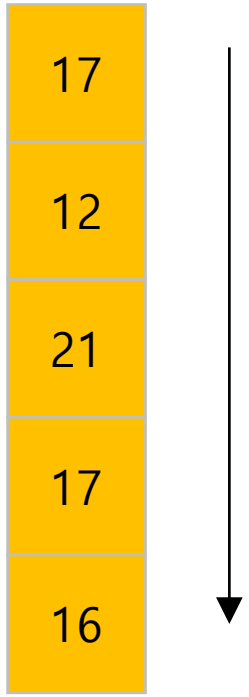
1	8	8	9	17
1	7	7	12	12
1	2	11	11	21
8	8	15	17	17
7	7	10	10	16

→

17
12
21
17
16

Step1 – 접근: 어떤 **줄**을 선택해야 할까?

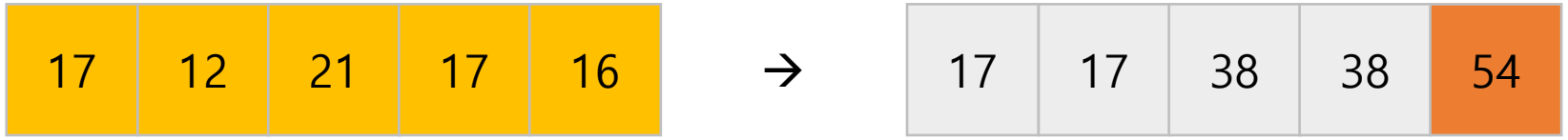
앞서서 각 줄에서 얻을 수 있는 최대 사탕 수를 구했음!



→ dp[] : 각 줄에 도달할 때까지 주울 수 있는 최대 사탕 수

→ 각 줄에 도달하는 방법
1. 두 번째 전 줄을 선택하고 자신의 줄을 선택
2. 바로 직전 줄을 선택하고 자신의 줄은 건너뜀

→ 각 경우 얻을 수 있는 사탕 수를 비교하여 최대 값이 자신의 dp값이 되도록 함



Step2 – 구현

```python
def one_line_max(arr, length):
    if length < 3:
        return max(arr)

    dp = [0]*length
    dp[0], dp[1] = arr[0], max(arr[0], arr[1])

    for i in range(2, length):
        dp[i] = max(dp[i-2] + arr[i], dp[i-1])

    print(dp)

    return dp[-1]
```

```python
while True:
    m, n = map(int, stdin.readline().split())

    if m == 0 and n == 0:
        break

    line_max = list()

    for i in range(m):
        line = list(map(int, stdin.readline().split()))
        line_max.append(one_line_max(line, n))

    print(one_line_max(line_max, m))
```

Thank you~!@#$%^&*()_+