



메기 농장

부 덩클렉은 메기 양어장을 가지고 있다. 양어장은 $N \times N$ 격자칸 모양이다. 격자의 칸들은 같은 크기의 정사각형이다. 격자의 열들은 서쪽에서 동쪽으로 0부터 $N - 1$ 까지 번호가 붙어 있고, 행들은 남쪽에서 북쪽으로 0부터 $N - 1$ 까지 번호가 붙어있다. 열 c , 행 r 에 ($0 \leq c \leq N - 1, 0 \leq r \leq N - 1$) 있는 칸을 칸 (c, r) 로 부른다.

양어장에는 M 마리의 메기가 있다. 메기들은 0부터 $M - 1$ 까지 번호가 붙어 있고, 모두 다른 칸에 있다. 각각의 i 에 대해 ($0 \leq i \leq M - 1$) 메기 i 는 칸 $(X[i], Y[i])$ 에 있고, 그 무게는 $W[i]$ 그램이다.

부 덩클렉은 메기를 잡기 위해 낚시터를 지으려고 한다. 열 c 에 있는 길이 k 인 낚시터는 ($0 \leq c \leq N - 1, 1 \leq k \leq N$), 열 c 의 행 0부터 행 $k - 1$ 까지를 덮는 직사각형이다. 즉, 낚시터는 칸들 $(c, 0), (c, 1), \dots, (c, k - 1)$ 를 덮는다. 각 열에 대해서 부 덩클렉은 특정한 길이의 낚시터를 짓거나, 낚시터를 전혀 짓지 않는 것 중 선택을 할 수 있다.

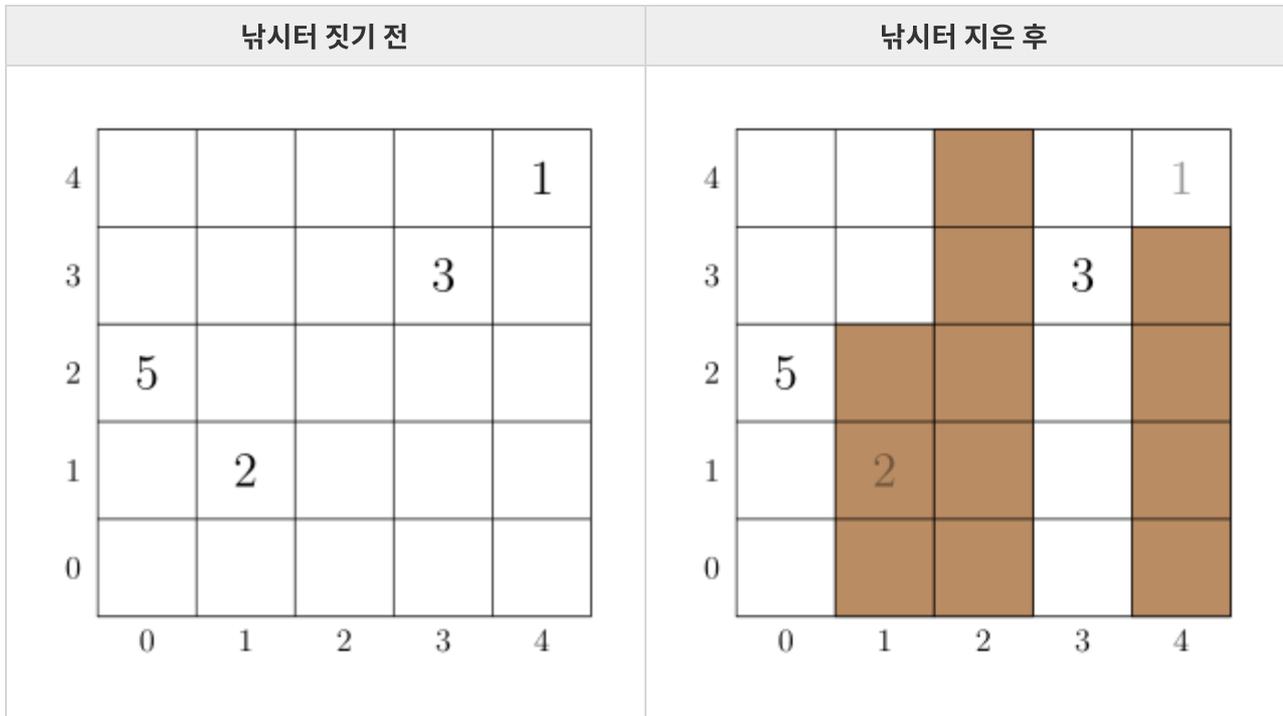
메기 i 를 ($0 \leq i \leq M - 1$) 잡기 위해서는 메기 i 의 위치의 서쪽이나 동쪽에 인접한 칸을 낚시터가 덮어야 하며, 메기 i 의 위치는 낚시터가 덮지 않아야 한다. 다시 말하면,

- 칸들 $(X[i] - 1, Y[i])$ 와 $(X[i] + 1, Y[i])$ 중 적어도 하나가 낚시터에 덮이고,
- 칸 $(X[i], Y[i])$ 는 낚시터에 덮이지 않아야 한다.

예를 들어, $N = 5$ 인 양어장에 $M = 4$ 마리의 메기가 있다고 하자.

- 메기 0의 위치는 칸 $(0, 2)$ 이고 그 무게는 5그램이다.
- 메기 1의 위치는 칸 $(1, 1)$ 이고 그 무게는 2그램이다.
- 메기 2의 위치는 칸 $(4, 4)$ 이고 그 무게는 1그램이다.
- 메기 3의 위치는 칸 $(3, 3)$ 이고 그 무게는 3그램이다.

부 덩클렉이 낚시터를 지을 수 있는 방법 중 하나는 아래와 같다.



칸에 표시된 자연수는 그 칸에 있는 메기의 무게이다. 색칠된 칸들이 낚시터에 덮인 곳이다. 이 경우 잡을 수 있는 메기는 메기 0(칸 (0, 2)에 위치)과 메기 3(칸 (3, 3)에 위치)이다. 메기 1(칸 (1, 1)에 위치)는 그 칸이 낚시터에 덮여 있어 잡을 수 없다. 메기 2(칸 (4, 4)에 위치)는 서쪽이나 동쪽에 인접한 칸이 낚시터로 덮인 것이 없어 잡을 수 없다.

부 뎅클렉은 잡을 수 있는 메기의 무게의 합이 가장 크도록 낚시터를 짓고 싶다. 잡을 수 있는 메기의 최대 무게 합을 계산하는 프로그램을 작성하라.

Implementation Details

당신은 다음 함수를 구현해야 한다:

```
int64 max_weights(int N, int M, int[] X, int[] Y, int[] W)
```

- N : 양식장의 크기.
- M : 메기의 마리 수.
- X, Y : 메기의 위치를 표현한 크기 M 인 배열들.
- W : 메기의 무게를 표현한 크기 M 인 배열.
- 이 함수는 부 뎅클렉이 낚시터를 지어서 잡을 수 있는 최대의 메기 무게 합을 리턴해야 한다.
- 이 함수는 정확히 한번 호출된다.

Example

다음 호출을 생각하자:

```
max_weights(5, 4, [0, 1, 4, 3], [2, 1, 4, 3], [5, 2, 1, 3])
```

이 예제는 위의 문제 설명에 있는 것과 같다.

문제 설명에 있는 것과 같이 낚시터를 지으면, 부 덩클렉은 메기 0과 3을 잡을 수 있고, 무게의 합은 $5 + 3 = 8$ 그램이다. 이 방법이 최선이며 함수는 8을 리턴해야 한다.

Constraints

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 300\,000$
- $0 \leq X[i] \leq N - 1, 0 \leq Y[i] \leq N - 1$ ($0 \leq i \leq M - 1$)
- $1 \leq W[i] \leq 10^9$ ($0 \leq i \leq M - 1$)
- 메기들의 위치는 모두 다르다. 즉, $X[i] \neq X[j]$ 혹은 $Y[i] \neq Y[j]$ ($0 \leq i < j \leq M - 1$).

Subtasks

1. (3 points) $X[i]$ 가 짝수 ($0 \leq i \leq M - 1$)
2. (6 points) $X[i] \leq 1$ ($0 \leq i \leq M - 1$)
3. (9 points) $Y[i] = 0$ ($0 \leq i \leq M - 1$)
4. (14 points) $N \leq 300, Y[i] \leq 8$ ($0 \leq i \leq M - 1$)
5. (21 points) $N \leq 300$
6. (17 points) $N \leq 3000$
7. (14 points) 각 열에는 최대 2 마리의 메기가 있다.
8. (16 points) 추가적인 제한이 없다.

Sample Grader

샘플 그레이더는 다음 형식으로 입력을 읽는다:

- line 1: $N M$
- line $2 + i$ ($0 \leq i \leq M - 1$): $X[i] Y[i] W[i]$

샘플 그레이더는 다음 형식으로 답을 출력한다:

- line 1: max_weights 의 리턴 값