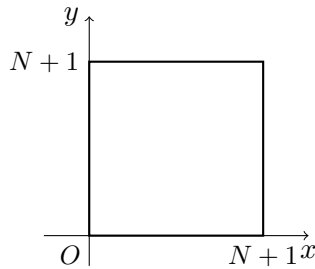


보안 시스템

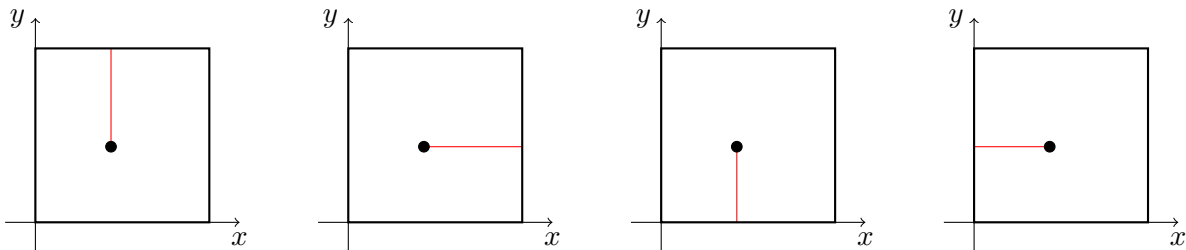
KOI국의 기밀 시설은 좌표평면 상에서 왼쪽 아래 꼭짓점이 $(0, 0)$, 오른쪽 위 꼭짓점이 $(N + 1, N + 1)$ 인 변들이 축에 평행한 정사각형으로 표현할 수 있다. 정사각형의 각 모서리는 기밀 시설들의 벽을 의미한다.



기밀 시설 내에는 N 개의 레이저 센서가 있다. 레이저 센서에는 0부터 $N - 1$ 까지의 번호가 붙어 있다. 기밀 시설의 침입자를 레이저 센서들을 통해 감지하는 보안 시스템을 설계하려고 한다.

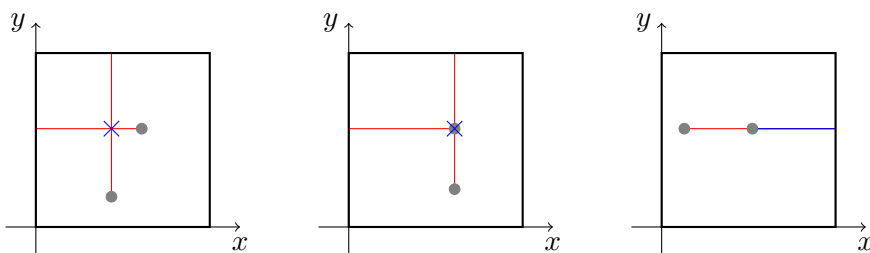
각각의 레이저 센서는 좌표평면 위의 점으로 나타낼 수 있다. 레이저 센서를 작동시킬 경우, 해당 위치에서 위쪽(+ y 축 방향), 오른쪽(+ x 축 방향), 아래쪽(- y 축 방향), 왼쪽(- x 축 방향) 중 하나의 방향으로 레이저 빛을 발사한다. 레이저 빛은 벽에 도달할 때까지 나아가므로, 레이저 빛의 경로는 레이저 센서가 있는 위치와 벽 위의 한 점을 잇는 선분으로 나타낼 수 있다.

레이저 빛이 발사되는 방향에는 1부터 4까지의 번호가 붙어 있다. 1은 위쪽 방향, 2는 오른쪽 방향, 3은 아래쪽 방향, 4는 왼쪽 방향을 나타낸다. 아래 그림은 1번, 2번, 3번, 4번 방향으로 레이저 빛을 발사하는 레이저 센서들의 예시를 순서대로 나타낸 것이다. 검은색 점은 레이저 센서를, 빨간색 선분은 레이저 빛을 나타낸다.



i ($0 \leq i \leq N - 1$)번 레이저 센서는 $(X[i], Y[i])$ 에 놓여 있고, 작동시킬 경우 빛을 $D[i]$ 번 방향으로 발사한다. 서로 다른 레이저 센서는 서로 다른 위치에 놓여 있다. $X[i]$ 와 $Y[i]$ 는 1 이상 N 이하의 정수이다.

각각의 레이저 센서의 작동 여부를 원하는 대로 정할 수 있다. 단, 서로 다른 레이저 센서가 발사한 레이저 빛이 만나면 인식 오류가 발생할 수 있기 때문에, 레이저 빛들이 끝 점을 포함해서 서로 만나지 않도록 해야 한다. 아래 그림은 레이저 빛끼리 만나는 예시로, 레이저 빛끼리는 한 점에서 만날 수도 있고, 선분으로 만날 수도 있다.



i ($0 \leq i \leq N - 1$)번 레이저 센서의 **중요도**는 $W[i]$ 이다. 중요도는, 해당 레이저 센서를 작동시켰을 때 보안에 얼마나 도움이 되는지를 수치적으로 나타낸 값이다. 전체 보안 시스템의 **보안 수준**은 작동시킨 레이저 센서들의 중요도 값의 합이다.

레이저 빔끼리 서로 만나지 않도록 레이저 센서들의 작동 여부를 결정했을 때, 가능한 보안 수준의 최댓값을 구하는 함수를 작성하라.

함수 목록 및 정의

여러분은 아래 함수를 구현해야 한다.

```
int max_level(vector<int> X, vector<int> Y, vector<int> D, vector<int> W)
```

- X, Y, D, W : 길이가 N 인 정수 배열. 모든 i ($0 \leq i \leq N - 1$)에 대해, i 번 레이저 센서의 좌표는 $(X[i], Y[i])$ 이고, 작동할 시 $D[i]$ 번 방향으로 빔을 발사하며, 중요도는 $W[i]$ 이다.
- 이 함수는 레이저 빔끼리 서로 만나지 않도록 레이저 센서들의 작동 여부를 결정했을 때, 가능한 보안 수준의 최댓값을 반환해야 한다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

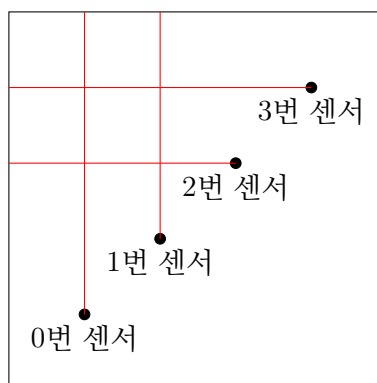
예제

$N = 4, X = [1, 2, 3, 4], Y = [1, 2, 3, 4], D = [1, 1, 4, 4], W = [1, 1, 1, 1]$ 인 경우를 생각해 보자.

그레이더는 다음과 같이 함수를 호출한다.

```
max_level({1, 2, 3, 4}, {1, 2, 3, 4}, {1, 1, 4, 4}, {1, 1, 1, 1})
```

아래 그림은 기밀 시설과 내부의 센서들, 그리고 센서들이 발사한 레이저 빔을 나타낸다. 0번 레이저 센서와 1번 레이저 센서를 작동시키거나, 2번 레이저 센서와 3번 레이저 센서를 작동시키면 레이저 빔이 서로 만나지 않으면서 보안 수준이 2가 된다. 이 두 방법보다 보안 수준을 더 크게 하는 방법은 존재하지 않는다.



따라서 함수는 2를 반환해야 한다.

제약 조건

- $1 \leq N \leq 1500$
- $1 \leq X[i], Y[i] \leq N$ (모든 $0 \leq i \leq N - 1$)
- $D[i] \in \{1, 2, 3, 4\}$ (모든 $0 \leq i \leq N - 1$)
- $1 \leq W[i] \leq 100\,000$ (모든 $0 \leq i \leq N - 1$)
- 각 센서는 모두 서로 다른 위치에 존재한다.

부분문제

1. (5점) $N \leq 18$
2. (8점) $N \leq 36$
3. (21점) $N \leq 100$
4. (15점) $N \leq 500$
5. (11점) $D[i] \in \{1, 2, 3\}$ (모든 $0 \leq i \leq N - 1$)
6. (17점) $X[i] \neq X[j]$ 이고 $Y[i] \neq Y[j]$ (모든 $0 \leq i < j \leq N - 1$)
7. (23점) 추가 제약 조건 없음

Sample grader

Sample grader의 입력 형식은 아래와 같다.

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

Sample grader의 출력 형식은 아래와 같다.

- Line 1: `max_level` 함수가 반환한 값

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.