분수 공원

공원에는 0부터 $n-1$로 번호가 매겨진 n개의 **분수**가 있다. 이 분수들을 2차원 평면에서 점들로 표현할 수 있다. 즉, 분수 i ($0 \le i \le n-1$)은 점 $(x[i], y[i])$으로 표현되는데, $x[i]$와 $y[i]$는 모두 **짝수인 정수**이다. 분수의 위치는 모두 다르다.

건축가 티모시는 몇 개의 **도로**와 도로마다 한 개의 **벤치**를 만들게 되었다. 도로는 **수평**이나 **수직**인 길이 2인 선분으로, 양 끝점은 서로 다른 두 분수이다. 어떤 두 분수도 도로를 따라서 이동하여 갈 수 있도록 도로를 놓아야 한다. 처음에는, 공원에 도로가 없다.

각각의 도로마다, **정확하게** 하나의 벤치를 놓을 것이며 이 도로에 **할당**된다. 각 벤치는 a와 b가 **모두 홀수**인 점 (a, b)에 놓여야 한다. 벤치의 위치는 모두 **달라야** 한다. (a, b)에 놓인 벤치는 도로의 **양 끝점 모두** $(a-1, b-1)$, $(a-1, b+1)$, $(a+1, b-1)$, $(a+1, b+1)$ 중 둘인 도로에 할당된다. 예를 들어서, $(3, 3)$에 놓인 벤치는 정확하게 하나의 도로에 할당될 수 있으며, 그 도로는 다음 네 선분 $(2, 2)$ – $(2, 4)$, $(2, 4)$ – $(4, 4)$, $(4, 4)$ – $(4, 2)$, $(4, 2)$ – $(2, 2)$ 중 하나이다.

티모시를 도와서, 위 모든 조건을 만족하게 도로를 만들고 벤치를 놓을 수 있는지 판단하자. 만약 둘 이상의 가능한 방법이 있다면, 이 중 어느 것을 보여줘도 좋다.

Implementation Details

다음 함수를 구현해야 한다.

```
int construct_roads(int[] x, int[] y)
```

- x, y: 길이 n인 두 배열. 각각의 i ($0 \le i \le n-1$)에 대해서, 분수 i는 점 $(x[i], y[i])$이고, $x[i]$와 $y[i]$는 짝수인 정수이다.
- 만약 조건을 만족하게 도로와 벤치를 놓을 수 있다면, 이 함수는 아래에 설명하는 `build` 함수를 정확히 한 번 호출하고 난 후 1을 리턴해야 한다.
- 그렇지 않으면, 이 함수는 `build`를 호출하지 않고 0을 리턴해야 한다.
- 이 함수는 정확하게 한 번 호출된다.

다음 함수를 호출하여 도로와 벤치를 놓는 가능한 방법을 제공할 수 있다.

```
void build(int[] u, int[] v, int[] a, int[] b)
```

- m이 만들어야 하는 도로의 개수라고 하자.
- u, v: 길이가 m인 두 배열로, 만들어야 하는 도로를 나타낸다. 이 도로는 0부터 $m-1$까지 번호가 매겨져 있다. 각각의 j에 대해서 ($0 \le j \le m-1$), 도로 j는 분수 $u[j]$와 $v[j]$를 연결한다. 각각의 도로는 수평 또는 수직인 길이 2인 선분이어야 한다. 서로 다른 두 도로는 최대 한 끝점(분수)를 공유할 수 있다. 일단 도로를 모두 만들고 나면, 어떤 두 분수도 도로를 통해서 연결되어야 한다.

- a, b: 길이 m인 두 배열로, 벤치를 나타낸다. 각각의 j에 대해서 ($0 \le j \le m - 1$), 벤치 하나가 $(a[j], b[j])$에 놓여지고, 도로 j에 할당된다. 둘 이상의 벤치가 같은 위치에 놓일 수 없다. 둘 이상의 벤치가 같은 도로에 할당될 수 없다.

Examples

Example 1

다음 호출을 생각해보자.

```
construct_roads([4, 4, 6, 4, 2], [4, 6, 4, 2, 4])
```

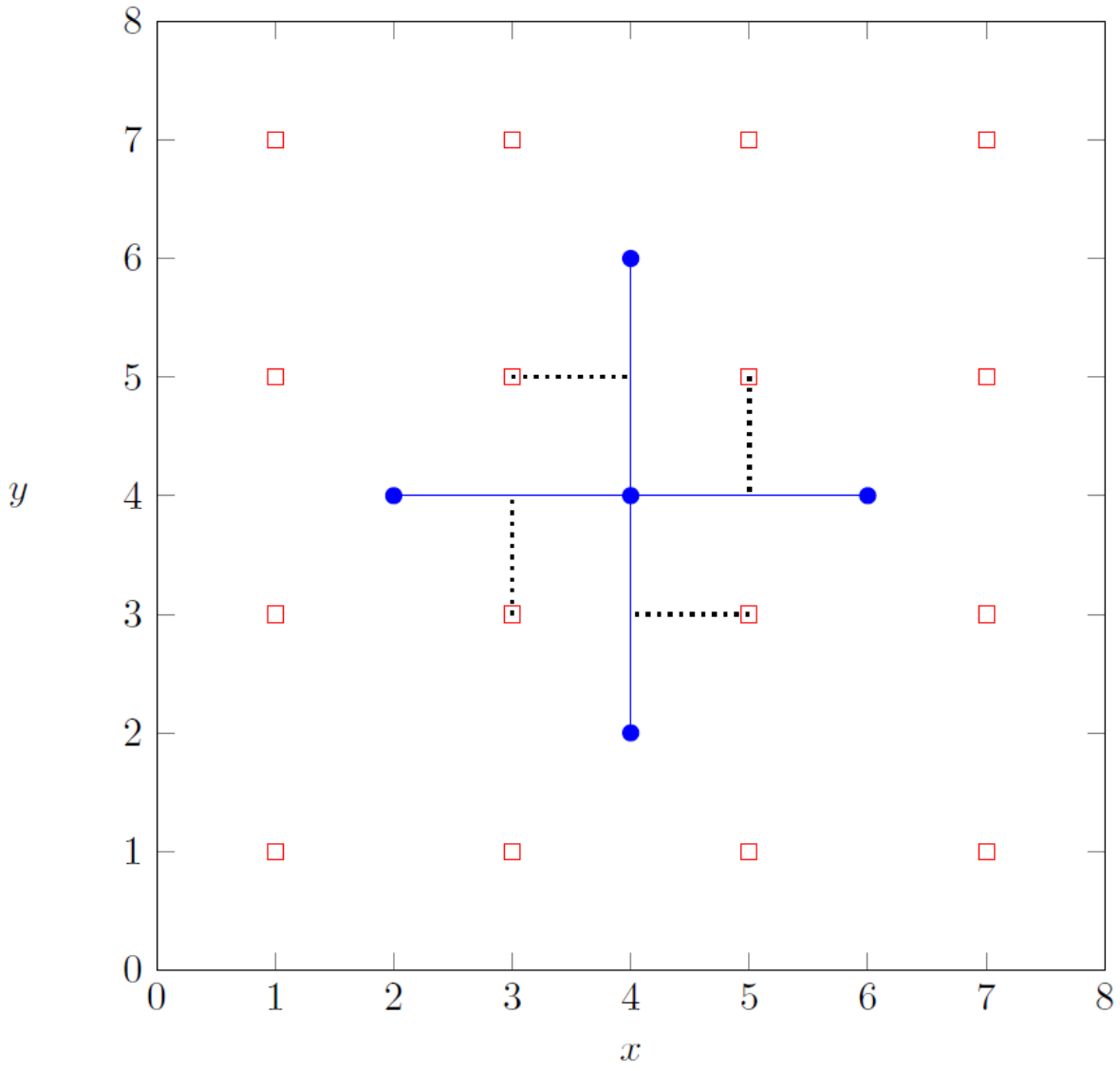
이는 다음과 같이 5개의 분수가 있다는 뜻이다.

- 분수 0이 $(4, 4)$,
- 분수 1이 $(4, 6)$,
- 분수 2가 $(6, 4)$,
- 분수 3이 $(4, 2)$,
- 분수 4가 $(2, 4)$.

다음과 같이 4개의 도로를 만들면, 각각의 도로가 두 분수를 연결하게 할 수 있고, 다음과 같이 벤치를 놓을 수 있다.

도로	도로가 연결하는 두 분수	할당된 벤치의 위치
0	$0, 2$	$(5, 5)$
1	$0, 1$	$(3, 5)$
2	$3, 0$	$(5, 3)$
3	$4, 0$	$(3, 3)$

위 답은 다음 그림으로 표현할 수 있다.



이 답을 보고하기 위해서, `construct_roads`는 다음과 같은 호출을 해야 한다.

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

이 함수의 리턴값은 1이어야 한다.

이 경우에, 주어진 조건을 만족하는 답은 여럿이 있고, 그 중 어느 것도 정답으로 인정된다. 예를 들어서, `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])`를 호출하고 1을 리턴해도 된다.

Example 2

다음 호출을 생각해보자.

```
construct_roads([2, 4], [2, 6])
```

분수 0가 $(2, 2)$에 있고 분수 1가 $(4, 6)$에 있다. 조건을 만족하도록 도로를 만들 수 있는 방법이 없기 때문에, `construct_roads`는 `build`를 호출하지 않고 0을 리턴해야 한다.

Constraints

- $1 \le n \le 200\,000$
- $2 \le x[i], y[i] \le 200\,000$ (모든 $0 \le i \le n-1$)
- $x[i]$와 $y[i]$ 는 짝수인 정수 (모든 $0 \le i \le n-1$).
- 둘 이상의 분수가 같은 위치에 있는 경우는 없다.

Subtasks

1. (5 점) $x[i] = 2$ (모든 $0 \le i \le n-1$)
2. (10 점) $2 \le x[i] \le 4$ (모든 $0 \le i \le n-1$)
3. (15 점) $2 \le x[i] \le 6$ (모든 $0 \le i \le n-1$)
4. (20 점) 어떤 두 분수도 도로를 따라서 이동할 수 있도록 도로를 놓는 방법이 최대 하나이다.
5. (20 점) 네 개의 분수가 2×2 크기 정사각형의 네 꼭지점을 이루는 경우는 없다.
6. (30 점) 추가적인 제약 조건이 없다.

Sample Grader

샘플 그레이더는 다음 양식으로 입력을 읽는다.

- line 1 : n
- line $2+i$ ($0 \le i \le n-1$): $x[i]$ $y[i]$

샘플 그레이더는 다음 양식으로 출력한다.

- line 1: `construct_roads`의 리턴 값

만약 `construct_roads` 의 리턴값이 1이고 `build(u, v, a, b)`가 호출되었다면, 그레이더는 추가로 다음을 출력한다.

- line 2: m
- line $3+j$ ($0 \le j \le m-1$): $u[j]$ $v[j]$ $a[j]$ $b[j]$