

噴水のある公園 (Fountain Parks)

二次元平面で表される近所の公園には n 個の噴水があり, 0 から $n - 1$ までの番号が付けられている. 噴水 i ($0 \leq i \leq n - 1$) は座標 $(x[i], y[i])$ に存在し, $x[i], y[i]$ は偶数である. また, すべての噴水の座標は相異なる.

建築家 Timothy はいくつかの道路の建設, そして各道路につき 1 つのベンチを割り当てる方法を計画する. 各道路は x 軸または y 軸に平行な長さ 2 の線分であり, 2 つの異なる噴水をつないでいる必要がある. また, どの 2 つの噴水の間もいくつかの道路をたどって移動できるようにする必要がある. はじめ, 道路は公園に 1 つも存在しない.

各道路につきちょうど 1 つのベンチを設置し, その道路に割り当てる必要がある. 各ベンチは a, b が奇数であるような座標 (a, b) に設置され, それらは相異なる座標になければならない. また, 座標 (a, b) のベンチは, 両方の端点が座標 $(a - 1, b - 1), (a - 1, b + 1), (a + 1, b - 1), (a + 1, b + 1)$ のうちいずれか 2 つである道路にしか割り当てることができない. 例えば, 座標 $(3, 3)$ に存在するベンチは,

- 線分 $(2, 2) - (2, 4)$ をなす道路
- 線分 $(2, 4) - (4, 4)$ をなす道路
- 線分 $(4, 4) - (4, 2)$ をなす道路
- 線分 $(4, 2) - (2, 2)$ をなす道路

のうちいずれかにしか割り当てることができない.

Timothy を助けるために, 前述の条件をすべて満たすような道路の建設とベンチの割り当てが可能かどうかを判定し, 可能な場合は条件を満たす答えを 1 つ報告しなさい. もし複数の答えが存在する場合はどれを報告してもよい.

実装上の注意

あなたは以下の関数を実装しなさい.

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

- x, y : それぞれ長さ n の配列である. それぞれの i ($0 \leq i \leq n - 1$) について, 噴水 i は座標 $(x[i], y[i])$ に存在する. $x[i], y[i]$ は偶数である.
- もし道路の建設とベンチの割り当てが可能の場合, この関数はちょうど 1 回 `build` (後述) を呼び出すことで答えを報告し, その後 1 を返す必要がある.
- それ以外の場合, この関数は `build` を呼び出さずに 0 を返す必要がある.
- この関数はちょうど 1 回呼び出される.

あなたのプログラムは, 適切な答えを報告するために, 以下の関数を呼び出すことができる.

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

- ここでは m を建設する道路の本数とする。
- u, v : それぞれ長さ m の配列であり、建設する道路の情報を表す。道路は 0 から $m - 1$ までの番号が付けられており、各 j ($0 \leq j \leq m - 1$) について、道路 j は噴水 $u[j]$ と噴水 $v[j]$ をつないでいる。各道路は x 軸または y 軸に平行な長さ 2 の線分である必要がある。どの 2 つの異なる道路も一方の端点以外で交わってはならない。また、道路が建設された後、どの 2 つの噴水の間もいくつかの道路をたどって移動できるようにする必要がある。
- a, b : それぞれ長さ m の配列であり、割り当てるベンチの位置の情報を表す。各 j ($0 \leq j \leq m - 1$) について、道路 j に割り当てられているベンチの座標は $(a[j], b[j])$ である。どの 2 つのベンチも異なる座標でなければならない。

入出力例

入出力例 1

以下の関数呼び出しを考える。

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

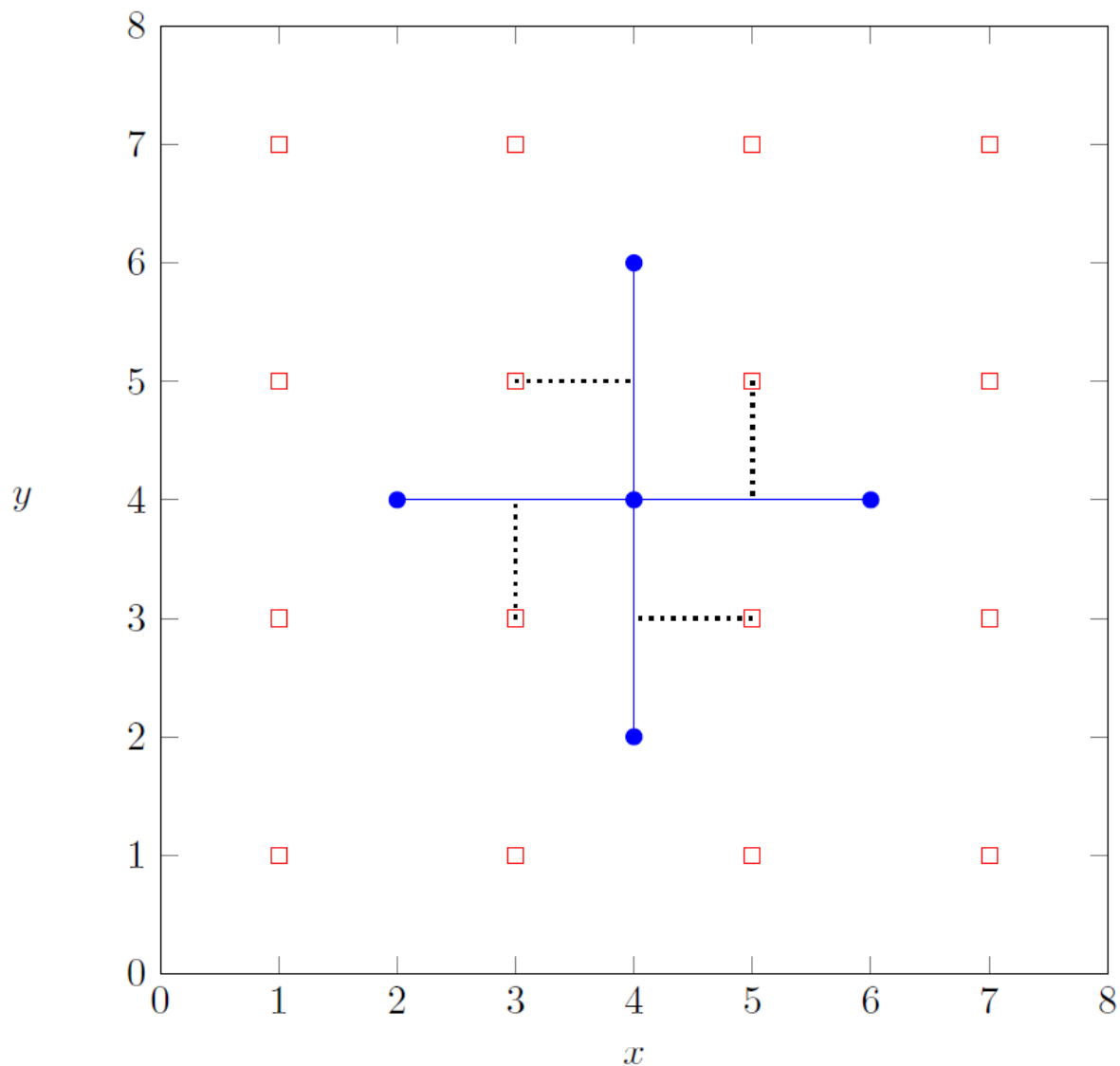
これは以下の 5 つの噴水があることを意味している。

- 噴水 0 が座標 $(4, 4)$ に存在する。
- 噴水 1 が座標 $(4, 6)$ に存在する。
- 噴水 2 が座標 $(6, 4)$ に存在する。
- 噴水 3 が座標 $(4, 2)$ に存在する。
- 噴水 4 が座標 $(2, 4)$ に存在する。

以下のように 4 つの道路を建設し、ベンチを各道路に割り当てると、問題文の条件を満たす。

道路の番号	道路で直接つながれている噴水の番号	ベンチの座標
0	0, 2	(5, 5)
1	0, 1	(3, 5)
2	3, 0	(5, 3)
3	4, 0	(3, 3)

この答えは、以下の図に対応する。



この答えを報告するためには、関数 `construct_roads` は以下の呼び出しを行う必要がある。

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

その後、関数 `construct_roads` は `1` を返さなければならない。

なお、この入力例では、問題文の条件を満たす答えが複数存在するが、そのうちどれを出力しても正解となる。例えば、`build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` を呼び出した後 `1` を返しても正解として扱われる。

入出力例 2

以下の関数呼び出しを考える。

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

噴水 0 は座標 $(2, 2)$ に存在し、噴水 1 は座標 $(4, 6)$ に存在する。問題文の条件を満たすような道路の建設方法は存在しないため、`construct_roads` は `build` を呼び出さずに `0` を返さなければならない。

制約

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$ ($0 \leq i \leq n - 1$)
- $x[i], y[i]$ は偶数である ($0 \leq i \leq n - 1$)
- どの 2 つの噴水も同じ座標に存在しない。

小課題

1. (5 点) $x[i] = 2$ ($0 \leq i \leq n - 1$)
2. (10 点) $2 \leq x[i] \leq 4$ ($0 \leq i \leq n - 1$)
3. (15 点) $2 \leq x[i] \leq 6$ ($0 \leq i \leq n - 1$)
4. (20 点) どの 2 つの噴水の間もいくつかの道路をたどって移動できるような道路の構築方法は、高々 1 通りしか存在しない。
5. (20 点) 大きさが 2×2 の正方形の頂点をなすような 4 つの噴水の組は存在しない。
6. (30 点) 追加の制約はない。

採点プログラムのサンプル

採点プログラムのサンプルは以下の形式で入力を読み込む。

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

採点プログラムのサンプルは以下の形式であなたの答えを出力する。

- 1 行目: `construct_roads` の戻り値。

さらに、`construct_roads` の戻り値が 1 であり、`build(u, v, a, b)` が呼び出された場合、以下の形式で追加で出力する。

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