

Fountain Parks

Một công viên có n **đài phun nước**, được đánh số từ 0 đến $n - 1$. Chúng ta mô hình hóa các đài phun nước là các điểm trên mặt phẳng tọa độ hai chiều. Đài phun nước thứ i ($0 \leq i \leq n - 1$) là điểm $(x[i], y[i])$ trong đó $x[i]$ và $y[i]$ là **các số nguyên chẵn**. Vị trí của các đài phun nước là đôi một khác nhau.

Timothy là kiến trúc sư được thuê để thiết kế xây dựng một số **con đường** và đặt một **ghế đá** trên mỗi con đường. Mỗi con đường là một đoạn thẳng **ngang** hoặc **dọc** với độ dài là 2 mà hai đầu mút là hai đài phun nước khác nhau. Các con đường cần được xây dựng sao cho có thể đi lại giữa hai đài phun nước bất kỳ bằng cách di chuyển trên các con đường. Ban đầu, không có con đường nào trong công viên.

Với mỗi con đường, có **đúng một** ghế đá cần phải được đặt trong công viên và được **chỉ định thuộc về** (nghĩa là quay mặt về phía) con đường đó. Mỗi ghế đá cần được đặt ở vị trí (a, b) sao cho a và b là **các số nguyên lẻ**. Vị trí của các ghế đá phải **đôi một khác nhau**. Một ghế đá tại vị trí (a, b) chỉ có thể thuộc về một con đường nếu **cả hai đầu** của con đường đó đều thuộc tập hợp các điểm $(a - 1, b - 1)$, $(a - 1, b + 1)$, $(a + 1, b - 1)$ và $(a + 1, b + 1)$. Ví dụ, ghế đá tại điểm $(3, 3)$ chỉ có thể thuộc về một con đường là một trong bốn đoạn thẳng sau $(2, 2) - (2, 4)$, $(2, 4) - (4, 4)$, $(4, 4) - (4, 2)$, $(4, 2) - (2, 2)$.

Hãy giúp Timothy xác định xem có cách xây dựng các con đường và đặt các ghế đá thỏa mãn tất cả các điều kiện ở trên hay không, nếu có, hãy giúp chỉ ra một phương án chấp nhận được. Nếu có nhiều phương án chấp nhận được thỏa mãn tất cả các điều kiện, bạn có thể đưa ra bất kỳ phương án nào.

Chi tiết cài đặt

Bạn cần cài đặt hàm sau:

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

- x, y : hai mảng độ dài n . Với mỗi i ($0 \leq i \leq n - 1$), đài phun nước i là điểm $(x[i], y[i])$, trong đó $x[i]$ và $y[i]$ là các số chẵn.
- Nếu có thể xây dựng được, hàm này cần gọi hàm `build` (xem ở dưới) đúng một lần để đưa ra phương án xây dựng, sau đó hàm này cần trả về `1`.
- Nếu không, hàm này cần trả về `0` mà không gọi hàm `build`.
- Hàm này chỉ được gọi đúng một lần.

Chương trình của bạn cần gọi hàm sau để đưa ra một phương án xây đường chấp nhận được và đặt vị trí các ghế đá:

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

- Gọi m là tổng số con đường sẽ được xây dựng.
- u, v : hai mảng độ dài m , thể hiện các con đường sẽ được xây dựng. Các con đường này được đánh số từ 0 đến $m - 1$. Với mỗi j ($0 \leq j \leq m - 1$), con đường j nối đài phun nước $u[j]$ và $v[j]$. Mỗi con đường phải là một đoạn thẳng ngang hoặc dọc với độ dài 2. Hai con đường khác nhau có thể có nhiều nhất một điểm chung (là một đài phun nước). Sau khi tất cả các con đường được xây dựng xong, chúng ta có thể di chuyển giữa hai đài phun nước bất kỳ thông qua các con đường.
- a, b : hai mảng độ dài m , thể hiện các ghế đá. Với mỗi j ($0 \leq j \leq m - 1$), một ghế đá được đặt tại vị trí $(a[j], b[j])$, và thuộc về con đường j . Không có hai ghế đá khác nhau nào có cùng vị trí.

Các ví dụ

Ví dụ 1

Xét lời gọi hàm sau:

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

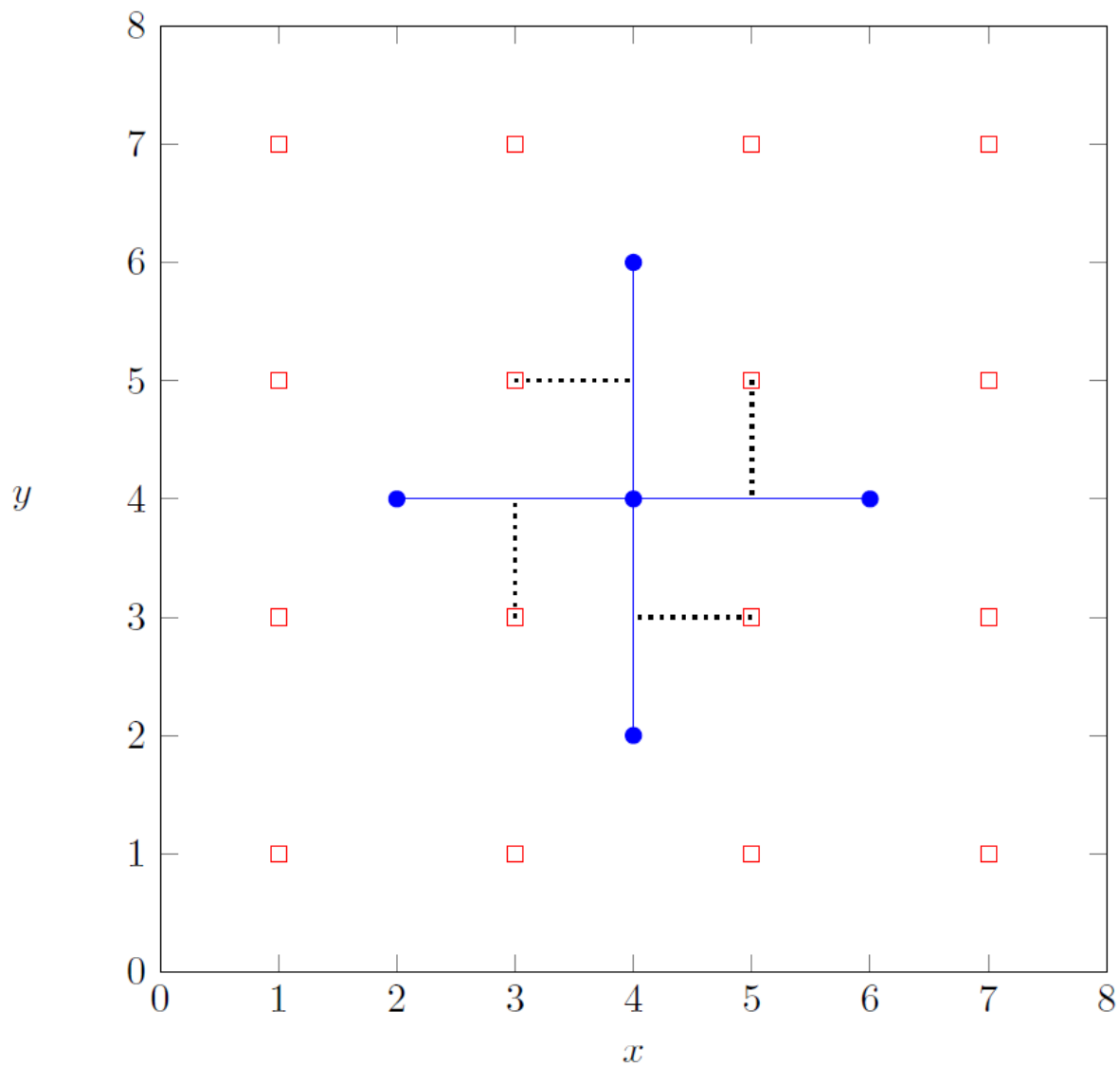
Với ý nghĩa là có 5 đài phun nước:

- đài phun nước 0 ở vị trí $(4, 4)$,
- đài phun nước 1 ở vị trí $(4, 6)$,
- đài phun nước 2 ở vị trí $(6, 4)$,
- đài phun nước 3 ở vị trí $(4, 2)$,
- đài phun nước 4 ở vị trí $(2, 4)$.

Có thể xây dựng 4 con đường, mỗi con đường nối hai đài phun nước, và đặt các ghế đá tương ứng như sau:

Nhãn của con đường	Nhãn của đài phun nước được nối với con đường	Vị trí ghế đá
0	0, 2	$(5, 5)$
1	0, 1	$(3, 5)$
2	3, 0	$(5, 3)$
3	4, 0	$(3, 3)$

Phương án này có thể được mô tả bằng hình vẽ sau:



Để đưa ra phương án này, `construct_roads` cần gọi hàm sau:

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

Hàm này cần trả về giá trị 1.

Lưu ý rằng trong trường hợp này, có nhiều phương án thỏa mãn các điều kiện và đều được coi là đúng. Ví dụ, phương án vẫn được coi là đúng nếu gọi `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` và trả về giá trị 1.

Ví dụ 2

Xét lời gọi hàm sau:

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

Đài phun nước 0 được đặt ở vị trí (2, 2) và đài phun nước 1 được đặt ở vị trí (4, 6). Vì không có phương án xây dựng các con đường thỏa mãn các điều kiện, `construct_roads` cần trả về 0 mà

không gọi hàm `build`.

Các ràng buộc

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$ (với mọi $0 \leq i \leq n - 1$)
- $x[i]$ và $y[i]$ là các số nguyên chẵn (với mọi $0 \leq i \leq n - 1$).
- Không có hai đài phun nước nào có cùng vị trí.

Các subtask

1. (5 điểm) $x[i] = 2$ (với mọi $0 \leq i \leq n - 1$)
2. (10 điểm) $2 \leq x[i] \leq 4$ (với mọi $0 \leq i \leq n - 1$)
3. (15 điểm) $2 \leq x[i] \leq 6$ (với mọi $0 \leq i \leq n - 1$)
4. (20 điểm) Chỉ có nhiều nhất một phương án xây dựng các con đường sao cho có thể di chuyển giữa hai đài phun nước bất kỳ bằng các con đường đó.
5. (20 điểm) Không tồn tại bốn đài phun nước nào tạo thành một hình vuông kích thước 2×2 .
6. (30 điểm) Không có ràng buộc nào thêm.

Trình chấm mẫu

Trình chấm mẫu đọc dữ liệu vào theo định dạng sau

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

Trình chấm mẫu đưa ra kết quả theo định dạng sau:

- dòng 1: giá trị trả về của `construct_roads`

Nếu kết quả trả về của hàm `construct_roads` là 1 và hàm `build(u, v, a, b)` có được gọi, trình chấm mẫu sẽ in ra thêm:

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