

噴泉公園 (Fountain Parks)

在附近的公園裡有 n 座噴泉，由 0 編號至 $n - 1$ 。我們視這些噴泉為二維平面上的點。噴泉 i ($0 \leq i \leq n - 1$) 為點 $(x[i], y[i])$ ，其中 $x[i]$ 和 $y[i]$ 為偶數。所有噴泉的位置皆不同。

建築師 Timothy 被雇用來規劃建造一些道路以及對於每條路擺放一張長椅。一條道路是一水平或垂直的線段，長度為 2 ，端點為兩個不同的噴泉。建造好的道路需使得任二噴泉間皆能沿著若干條道路通往彼此。初始時，公園中沒有任何道路。

對每條道路，恰好一張長椅需被擺放在公園中，稱該長椅被指定給（即，面向）該道路。每張長椅必須被擺放在某個點 (a, b) 滿足 a 與 b 皆為奇數。所有長椅的位置都不相同。一張位於 (a, b) 的長椅僅能被指定給兩端點落於 $(a - 1, b - 1)$ 、 $(a - 1, b + 1)$ 、 $(a + 1, b - 1)$ 與 $(a + 1, b + 1)$ 這些點之中的某一條道路。舉例來說，位於 $(3, 3)$ 的長椅僅可被指定給下面這四條路之一 $(2, 2) - (2, 4)$ 、 $(2, 4) - (4, 4)$ 、 $(4, 4) - (4, 2)$ 、 $(4, 2) - (2, 2)$ 。

請幫助 Timothy 判斷是否有可能建造道路並擺放長椅來滿足前述的條件。若所求存在，請提供一個可行的解。若有多個可行解，回報任一個即可。

實作細節 (Implementation Details)

你應實作下列函式：

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

- x, y : 兩個長度為 n 的陣列。對每個 i ($0 \leq i \leq n - 1$)，噴泉 i 為點 $(x[i], y[i])$ ，其中 $x[i]$ 與 $y[i]$ 都是偶數。
- 若所求的建造方式是可能的，此函式應呼叫 build (如下) 恰好一次，來回報一解，然後再回傳 (return) 1 。
- 否則，此函式應回傳 0 ，且不呼叫 build。
- 此函式被呼叫恰好一次。

你的實作可以呼叫下列函式來回報一可行的道路建造與長椅擺放方式：

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

- 令 m 是所建造的道路數目。
- u, v : 兩個長度為 m 的陣列，表示要建造的道路。這些道路由 0 編號至 $m - 1$ 。對每個 j ($0 \leq j \leq m - 1$)，道路 j 連接噴泉 $u[j]$ 和 $v[j]$ 。每條道路必須為水平或垂直，且長度為 2 的線段。任二相異的道路可有至多一個相同點（即一噴泉）。一旦道路建造完成，任二噴泉間皆能沿著若干條道路通往彼此。

- a, b : 兩個長度為 m 的陣列，表示長椅。對每個 j ($0 \leq j \leq m - 1$)，有一張長椅被擺放於 $(a[j], b[j])$ ，且被指定給道路 j 。兩張不同的長椅位置須相異。不同的長椅不可被指定給同一條道路。

範例

範例 1

考慮下列呼叫：

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

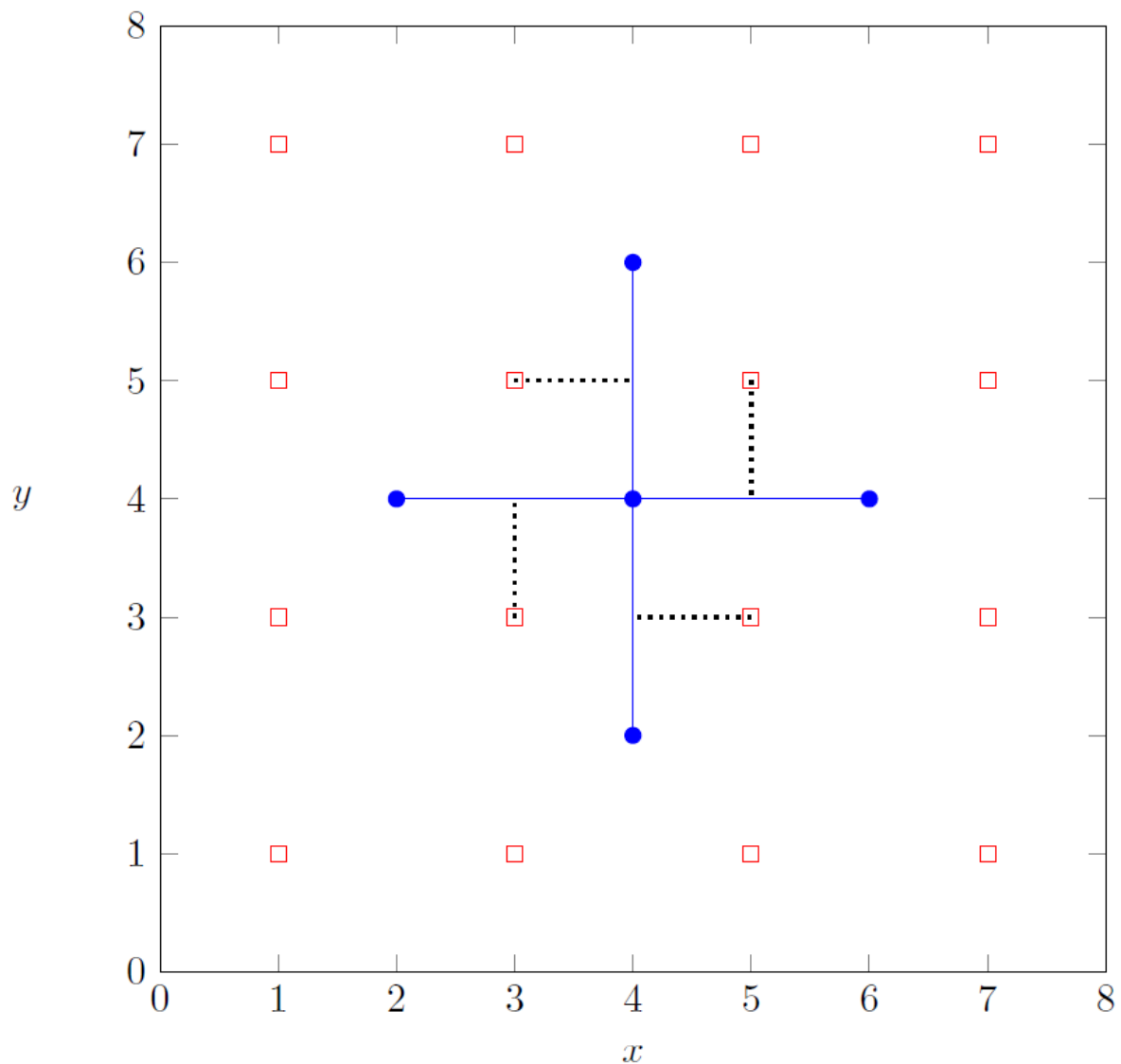
這表示有 5 座噴泉：

- 噴泉 0 位於 $(4, 4)$,
- 噴泉 1 位於 $(4, 6)$,
- 噴泉 2 位於 $(6, 4)$,
- 噴泉 3 位於 $(4, 2)$,
- 噴泉 4 位於 $(2, 4)$.

可以建造以下 4 條道路，每條連接兩個噴泉且可擺放對應的長椅：

道路編號	該道路連接的兩個噴泉編號	被指定的長椅位置
0	0, 2	$(5, 5)$
1	0, 1	$(3, 5)$
2	3, 0	$(5, 3)$
3	4, 0	$(3, 3)$

此解對應於下面的圖：



要回報此解，`construct_roads` 應呼叫：

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

然後回傳 `1`。

注意在此例中，有多個滿足要求的解，所有皆被視為正確。例如，呼叫 `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` 然後回傳 `1` 也是正確的。

範例 2

考慮下列呼叫：

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

噴泉 `0` 位於 `(2,2)` 且噴泉 `1` 位於 `(4,6)`。因為不存在滿足要求的建造方式，`construct_roads` 應回傳 `0` 且不呼叫 `build`。

限制 (Constraints)

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$ (對所有 $0 \leq i \leq n - 1$)
- $x[i]$ 與 $y[i]$ 皆為偶數 (對所有 $0 \leq i \leq n - 1$)
- 任兩座噴泉位置皆相異。

子任務 (Subtasks)

1. (5 points) $x[i] = 2$ (對所有 $0 \leq i \leq n - 1$)
2. (10 points) $2 \leq x[i] \leq 4$ (對所有 $0 \leq i \leq n - 1$)
3. (15 points) $2 \leq x[i] \leq 6$ (對所有 $0 \leq i \leq n - 1$)
4. (20 points) 至多只有一種建造道路的方式，使得任兩噴泉間沿著若干條道路能夠通往彼此。
5. (20 points) 任意四個噴泉都不會形成 2×2 正方形的頂點。
6. (30 points) 無額外限制。

範例評分程式 (Sample Grader)

此範例評分程式以下列格式讀取輸入：

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

此範例評分程式以下列格式輸出：

- line 1: `construct_roads` 的回傳值

若 `construct_roads` 的回傳值為 1 且 `build(u, v, a, b)` 有被呼叫，此評分程式會再輸出：

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