Парк фонтанів

У найближчому парку є n **фонтанів**, що пронумеровано від 0 до $n-1$. Ми позначаємо фонтани точками на площині, а саме, фонтан i ($0 \leq i \leq n-1$) є точкою $(x[i], y[i])$, де $x[i]$ та $y[i]$ є **парними цілими**. Усі фонтани розташовано у різних місцях.

Архітектора Тимофія запросили щоб спланувати будівництво деяких **доріг** та розташування однієї **лави** на кожній дорозі. Дорога є **горизонтальним** або **вертикальним** відрізком довжини 2, на кінцях якого розташовано два різних фонтана. Дороги треба збудувати так, щоб можна було подорожувати між довільними двома фонтанами пересуваючись по дорогах. На початку у парку немає доріг.

На кожній дорозі треба встановити **рівно** одну лаву, яка **відноситься до** (тобто, розташована передом до) цієї дороги. Кожна лава має бути розташована у деякій точці (a, b), так що a та b є **непарними цілими**. Позиції усіх лав мають бути **різними**. Лава у позиції (a, b) може відноситись до дороги тільки якщо **обидва** кінця дороги є серед $(a-1, b-1)$, $(a-1, b+1)$, $(a+1, b-1)$ та $(a+1, b+1)$. Наприклад, лава у позиції $(3, 3)$ може відноситись до дороги, яка є одним з чотирьох відрізків $(2, 2)$ – $(2, 4)$, $(2, 4)$ – $(4, 4)$, $(4, 4)$ – $(4, 2)$, $(4, 2)$ – $(2, 2)$.

Допоможіть Тимофію визначити, чи можна збудувати дороги та розставити лави так, щоб задовольнялись усі умови, зазначені вище, і якщо так, надайте йому прийнятний розв'язок. Якщо є кілька прийнятних розв'язків що задовольняють усім умовам, ви можете надати довільний з них.

Деталі реалізації

Ви маєте реалізувати наступну процедуру:

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

- x, y: Два масиви довжини n. Для кожного i ($0 \leq i \leq n-1$), фонтан i це точка $(x[i], y[i])$, де $x[i]$ та $y[i]$ є парними цілими.
- Якщо є можливість будівництва, ця процедура має зробити рівно один виклик `build` (дивись нижче) щоб повідомити розв'язок, після чого вона має повернути 1.
- Інакше, процедура має повернути 0 не викликаючи `build`.
- Ця процедупа викликається рівно один раз.

Ваша реалізація має викликати наступну процедуру щоб повідомити прийнятний план будівництва доріг та розташування лав:

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

- Нехай m буде загальною кількістю побудованих доріг.
- u, v: два масиви довжини m, що задають дороги які мають бути збудовані. Дороги пронумеровано від 0 до $m - 1$. Для кожного j ($0 \leq j \leq m - 1$), дорога j з'єднує фонтани $u[j]$ та $v[j]$. Кожна дорога має бути горизонтальним або вертикальним відрізком довжини 2. Дві довільні різні дороги можуть мати не більше однієї спільної точки (фонтан). Як тільки дороги збудовано, має існувати можливість подорожувати між довільними двома фонтанами рухаючись по дорогах.
- a, b:два масиви довжини m, що задають лави. Для кожного j ($0 \leq j \leq m - 1$), лаву розташовано уточці $(a[j], b[j])$, та вона відноситься до дороги j. Ніякі дві різні лави не можуть знаходитись у однаковій позиції.

Приклади

Приклад 1

Позглянемо наступний виклик:

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

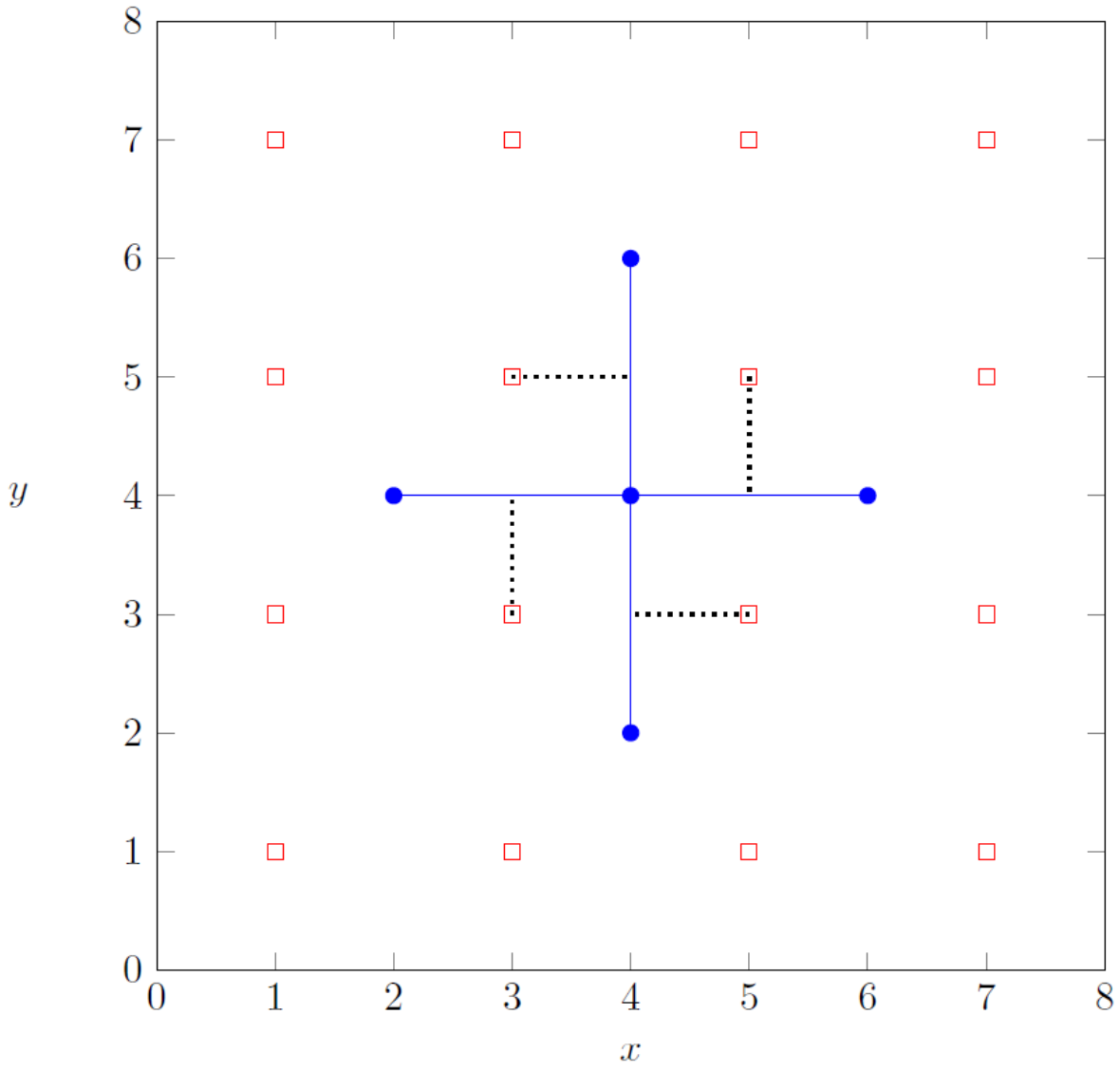
Це одначає, що є 5 фонтанів:

- фонтан 0 розташовано у $(4, 4)$,
- фонтан 1 розташовано у $(4, 6)$,
- фонтан 2 розташовано у $(6, 4)$,
- фонтан 3 розташовано у $(4, 2)$,
- фонтан 4 розташовано у $(2, 4)$.

Можна збудувати наступні 4 дороги, де кожна дорога сполучає два фонтани, та розташувати відповідні лави:

Мітка дорогиI	Мітки фонтанів, що сполучає дорога	Розташувння відповідної лави
0	$0, 2$	$(5, 5)$
1	$0, 1$	$(3, 5)$
2	$3, 0$	$(5, 3)$
3	$4, 0$	$(3, 3)$

Цей розв'язок відповідає наступному рисунку:



Щоб повідомити розв'язок, `construct_roads` має зробити наступний виклик:

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

Після цього вона має повернути 1.

Зауважте, що у цьому прикладі є кілька розв'язків що задовольняють вимогам, усі вони будуть вважатись коректними. Наприклад, також вірним буде зробити виклик `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` та потім повернути 1.

Приклад 2

Розглянемо наступний виклик:

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

Фонтан 0 розташовано у $(2, 2)$ та фонтан 1 розташовано у $(4, 6)$. Оскільки немає можливості побудувати дороги що задовольняють вимогам, `construct_roads` має повернути 0 не

викликаючи `build`.

Обмеження

- $1 \le n \le 200\,000$
- $2 \le x[i], y[i] \le 200\,000$ (для всіх $0 \le i \le n - 1$)
- $x[i]$ та $y[i]$ є парними числами (для всіх $0 \le i \le n - 1$).
- Ніякі два фонтана не розташовано у тій самій позиції.

Підзадачі

1. (5 балів) $x[i] = 2$ (для всіх $0 \le i \le n - 1$)
2. (10 балів) $2 \le x[i] \le 4$ (для всіх $0 \le i \le n - 1$)
3. (15 балів) $2 \le x[i] \le 6$ (для всіх $0 \le i \le n - 1$)
4. (20 балів) Є не більше одного способу побудувати дороги, так що можна полорожувати між довільними двома фонтанами рухаючись по дорогах.
5. (20 балів) Немає чтотирьох фонтанів, що утворюють кути квадрату 2×2.
6. (30 балів) Без додаткових обмежень.

Приклад модуля перевірки

Приклад модуля перевірки читає вхідні дані у наступному форматі:

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

Вивід модуля перевірки має такий формат:

- рядок 1: значення, що повернув виклик `construct_roads`

Якщо виклик `construct_roads` повернув 1 та був виклик `build(u, v, a, b)`, модуль перевірки додатково друкує:

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