

Fontány v parku

V parku je n **fontán** očíslovaných od 0 do $n - 1$, které budeme reprezentovat jako body v rovině. Přesněji, fontána i ($0 \leq i \leq n - 1$) je bod $(x[i], y[i])$, kde $x[i]$ a $y[i]$ jsou **sudá přirozená čísla**. Pozice fontán jsou navzájem různé.

Stavař Tim má za úkol postavit **cesty** propojující fontány a ke každé z těchto cest umístit **lavičku**. Každá z cest je **horizontální** nebo **vertikální** úsečka délky 2, jejíž konce jsou fontány. Po těchto cestách musí být možné přejít mezi každými dvěma fontánami. Na začátku v parku žádné cesty nejsou.

U každé cesty musí být umístěna **právě** jedna lavička **přiřazená** (natočená) k této cestě. Lavičky mohou být umístěny pouze v **navzájem různých** bodech (a, b) takových, že a i b jsou **lichá přirozená čísla**. Lavička na pozici (a, b) může být přiřazena k cestě pouze tehdy, když **oba** konce této cesty jsou rovné jednomu z bodů $(a - 1, b - 1)$, $(a - 1, b + 1)$, $(a + 1, b - 1)$ a $(a + 1, b + 1)$. Například lavička $(3, 3)$ může být přiřazena pouze k jedné z cest reprezentovaných úsečkami $(2, 2) - (2, 4)$, $(2, 4) - (4, 4)$, $(4, 4) - (4, 2)$, nebo $(4, 2) - (2, 2)$.

Najděte způsob, jak postavit cesty a umístit k nim lavičky tak, aby byly splněny výše popsané podmínky, nebo rozhodněte, že to není možné. Existuje-li více řešení, můžete vrátit libovolné z nich.

Implementační detaily

Implementujte následující funkci:

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

- x, y : dvě pole délky n . Pro každé i ($0 \leq i \leq n - 1$) je fontána číslo i reprezentována bodem $(x[i], y[i])$, kde $x[i]$ a $y[i]$ jsou sudá přirozená čísla.
- Existuje-li alespoň jedno řešení, popište libovolné z nich pomocí právě jednoho volání funkce `build` popsané níže a vraťte 1.
- Jestliže řešení neexistuje, funkci `build` nevolejte a vraťte 0.
- Vyhodnocovač tuto funkci volá právě jednou.

Popis cest a umístění laviček vraťte voláním následující funkce:

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

- Označme symbolem m celkový počet postavených cest.
- u, v : dvě pole délky m popisující cesty očíslované od 0 do $m - 1$. Pro každé j ($0 \leq j \leq m - 1$), cesta číslo j spojuje fontány číslo $u[j]$ a $v[j]$. Každá cesta musí být

horizontální nebo vertikální úsečka délky 2. Pomocí těchto cest musí být možné přejít mezi každými dvěma fontánami,

- a, b : dvě pole délky m popisující lavičky. Pro každé j ($0 \leq j \leq m - 1$) je lavička umístěna na pozici $(a[j], b[j])$ a přiřazena k cestě číslo j . Žádné dvě lavičky nesmí být reprezentované stejným bodem.

Příklady

Příklad 1

Uvažme následující volání:

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

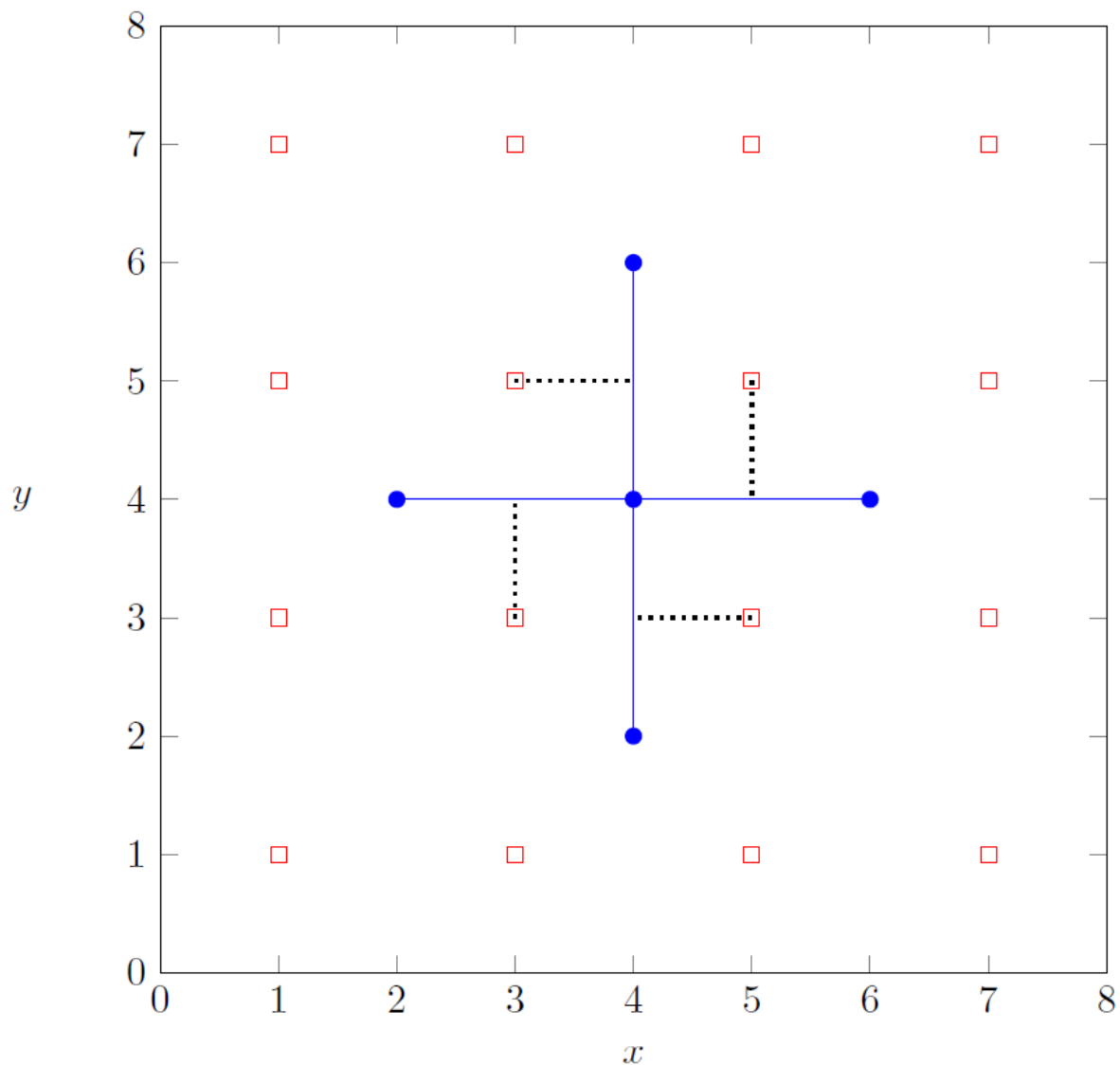
V parku je tedy 5 fontán:

- fontána 0 v bodě $(4, 4)$,
- fontána 1 v bodě $(4, 6)$,
- fontána 2 v bodě $(6, 4)$,
- fontána 3 v bodě $(4, 2)$,
- fontána 4 v bodě $(2, 4)$.

Řešením je postavit následující 4 cesty, z nichž každá spojuje dvojici fontán, a přiřadit k nim lavičky například takto:

Číslo cesty	Čísla fontán propojených touto cestou	Pozice přiřazené lavičky
0	0, 2	$(5, 5)$
1	0, 1	$(3, 5)$
2	3, 0	$(5, 3)$
3	4, 0	$(3, 3)$

Toto řešení je zobrazeno na následujícím obrázku:



Funkce `construct_roads` popíše toto řešení voláním

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

a poté vrátí 1.

Jiné možné správné řešení je zavolat `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` a poté vrátit 1.

Příklad 2

Uvažme následující volání:

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

Fontána 0 je umístěna v bodě (2,2) a fontána 1 je umístěna v bodě (4,6), nelze je tedy propojit cestou. Funkce `construct_roads` proto nevolá funkci `build` a vrátí 0.

Omezení

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$ (pro $0 \leq i \leq n - 1$)
- $x[i]$ a $y[i]$ jsou sudá přirozená čísla (pro $0 \leq i \leq n - 1$).
- Žádné dvě fontány nejsou reprezentované stejným bodem.

Podúlohy

1. (5 bodů) $x[i] = 2$ (pro $0 \leq i \leq n - 1$)
2. (10 bodů) $2 \leq x[i] \leq 4$ (pro $0 \leq i \leq n - 1$)
3. (15 bodů) $2 \leq x[i] \leq 6$ (pro $0 \leq i \leq n - 1$)
4. (20 bodů) Existuje nejvýše jeden způsob jak postavit cesty tak, aby bylo možné přejít mezi každými dvěma fontánami.
5. (20 bodů) Žádné čtyři fontány netvoří vrcholy čtverce o rozměrech 2×2 .
6. (30 bodů) Bez dalších omezení.

Ukázkový vyhodnocovač

Ukázkový vyhodnocovač čte vstup v následujícím formátu:

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

Výstup je vypsán v následujícím formátu:

- řádka 1: návratová hodnota funkce `construct_roads`

Je-li návratová hodnota funkce `construct_roads` 1 a je zavolána funkce `build(u, v, a, b)`, vyhodnocovač dále vypíše:

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