

Fontány (Fountain Parks)

V parku je n fontánok. Majú čísla od 0 po $n - 1$. Pre účely tejto úlohy sú fontány body vo vodorovnej dvojrozmernej rovine. Špecificky, fontánka i ($0 \leq i \leq n - 1$) má súradnice $(x[i], y[i])$, pričom $x[i]$ aj $y[i]$ sú **párne** celé čísla. Žiadne dve fontány neležia v tom istom bode.

V parku zatiaľ nie sú žiadne chodníky. Dľaždič Dezider preto dostal za úlohu spraviť v parku nejaké *chodníky* a ku každému chodníku aj jednu lavičku.

Chodník je **vodorovná alebo zvislá** úsečka dĺžky presne 2 , ktorá spája dve fontány. Chodníky je potrebné postaviť tak, aby sa po nich dalo dostať od každej fontány ku každej inej.

Pre každý chodník je potrebné postaviť aj **práve jednu** lavičku, ktorá bude **priradená** ku danému chodníku (t.j. otočená tak, aby si ľudia idúci po dotyčnom chodníku vedeli na ňu sadnúť). Každú lavičku treba umiestniť do nejakého bodu, ktorého obe súradnice sú **nepárne** celé čísla. Všetky lavičky musia byť **v navzájom rôznych** bodoch.

Lavičku v bode (a, b) môžeme priradiť konkrétnemu chodníku práve vtedy, keď **obe** súradnice koncov chodníka sú niektoré z bodov $(a - 1, b - 1)$, $(a - 1, b + 1)$, $(a + 1, b - 1)$ a $(a + 1, b + 1)$. Napríklad lavičku na súradniciach $(3, 3)$ môžeme priradiť len jednému zo štyroch možných chodníkov: $(2, 2) - (2, 4)$, $(2, 4) - (4, 4)$, $(4, 4) - (4, 2)$, $(4, 2) - (2, 2)$.

Pomôž Deziderovi. Zisti, či je možné podľa vyššie uvedených podmienok postaviť chodníky a rozmiestniť a priradiť im lavičky. Ak áno, nájdi ľubovoľné jedno platné riešenie.

Detaily implementácie

Tvojou úlohou je implementovať túto funkciu:

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

- x, y : dve polia dĺžky n . Pre každé i ($0 \leq i \leq n - 1$), fontánka i má súradnice $(x[i], y[i])$, kde $x[i]$ a $y[i]$ sú párne celé čísla.
- Ak sa dá všetko potrebné postaviť, tvoja funkcia by mala oznámiť riešenie graderu tým, že práve raz zavolá jeho funkciu `build` (popísaná nižšie). Následne by tvoja funkcia mala vrátiť hodnotu `1` a skončiť.
- Ak riešenie neexistuje, tvoja funkcia má vrátiť hodnotu `0` a skončiť bez toho, aby zavolala `build`.
- Túto funkciu grader zavolá práve raz.

Tvoja funkcia odovzdáva svoju konštrukciu chodníkov a lavičiek zavolaním nasledujúcej funkcie gradera:

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

- Nech m je celkový počet chodníkov, ktoré chceš postaviť.
- u, v : dve polia dĺžky m , popisujúce chodníky, ktoré chceš stavať. Chodníky dostanú čísla od 0 po $m - 1$. Pre každé j ($0 \leq j \leq m - 1$), chodník j spája fontánky s číslami $u[j]$ a $v[j]$. Pamätajte, že každý chodník musí byť vodorovný alebo zvislý a musí mať dĺžku práve 2. Každé dva chodníky musia mať najviac jeden spoločný bod (fontánku na konci). Po chodníkoch sa musí dať navštíviť všetky fontánky.
- a, b : dve polia dĺžky m popisujúce lavičky. Pre každé j ($0 \leq j \leq m - 1$) máme postaviť lavičku na súradniciach $(a[j], b[j])$ a priradiť ju ku chodníku j . Žiadne dve lavičky nesmú stáť v tom istom bode.

Príklady

Príklad 1

Uvažujme toto volanie tvojej funkcie:

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

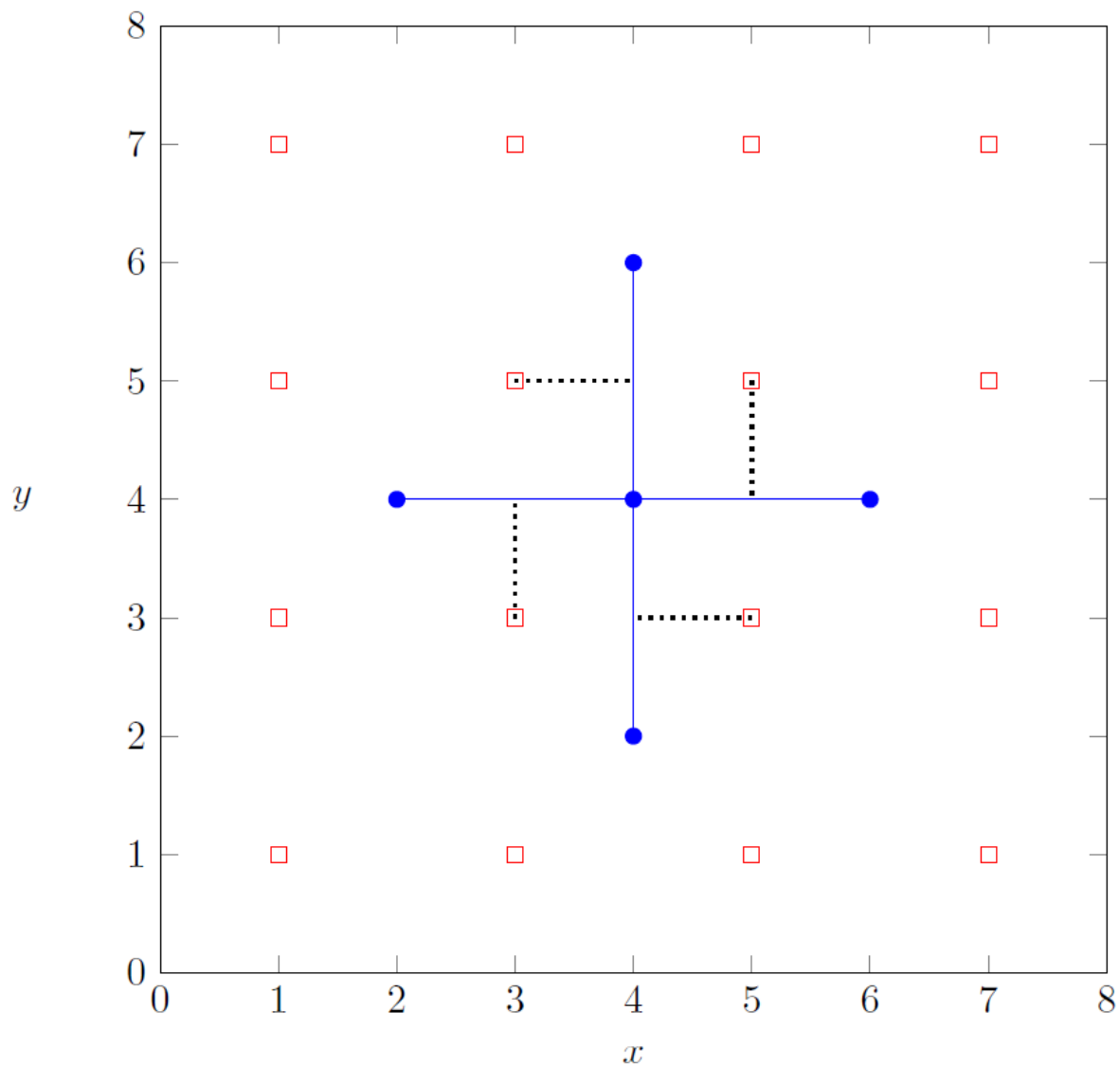
Máme teda 5 fontánok:

- fontánka 0 v bode $(4, 4)$,
- fontánka 1 v bode $(4, 6)$,
- fontánka 2 v bode $(6, 4)$,
- fontánka 3 v bode $(4, 2)$,
- fontánka 4 v bode $(2, 4)$.

Tieto fontánky vieme pospájať tak, že nižšie popísaným spôsobom postavíme štyri chodníky a štyri lavičky.

číslo chodníka	čísla fontánok, ktoré spája	poloha jemu priradenej lavičky
0	0, 2	$(5, 5)$
1	0, 1	$(3, 5)$
2	3, 0	$(5, 3)$
3	4, 0	$(3, 3)$

Toto riešenie na obrázku:



Ak by tvoja funkcia chcela odovzdať toto riešenie, mala by spraviť nasledovné volanie funkcie:

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

a následne skončiť s návratovou hodnotou `1`.

Toto riešenie nie je jediné možné. Tiež by napríklad bolo v poriadku, ak tvoja funkcia zavolá `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` a potom vráti `1`.

Príklad 2

Uvažujme toto volanie tvojej funkcie:

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

Fontánka `0` má súradnice `(2, 2)` a fontánka `1` zase `(4, 6)`. Keďže tieto dve fontánky nevieme povoleným spôsobom prepojiť chodníkmi, má tvoja funkcia vrátiť `0` a nikdy nezavolať `build`.

Obmedzenia

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$ (pre všetky $0 \leq i \leq n - 1$)
- $x[i]$ and $y[i]$ sú párne (pre všetky $0 \leq i \leq n - 1$).
- Žiadne dve fontánky nie sú v tom istom bode.

Podúlohy

1. (5 points) $x[i] = 2$ (pre všetky $0 \leq i \leq n - 1$)
2. (10 points) $2 \leq x[i] \leq 4$ (pre všetky $0 \leq i \leq n - 1$)
3. (15 points) $2 \leq x[i] \leq 6$ (pre všetky $0 \leq i \leq n - 1$)
4. (20 points) Existuje nanajvýš jeden spôsob, ako postaviť platnú sadu chodníkov tak, aby prepojila všetky fontánky.
5. (20 points) Žiadne štyri fontánky netvoria vrcholy štvorca rozmerov 2×2 .
6. (30 points) Bez ďalších obmedzení.

Ukážkový grader

Ukážkový grader očakáva vstup v nasledovnom formáte:

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

Ukážkový grader najskôr vypíše:

- riadok 1: návratová hodnota tvojej funkcie `construct_roads`

Ak tvoja funkcia vrátila 1 a naozaj pred tým zavolała `build(u, v, a, b)`, ukážkový grader následne vypíše:

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