

Fountain Parks

Park je uljepšan sa n **fontana** koje su označene brojevima od 0 do $n - 1$. Fontane možemo modelovati kao cjelobrojne tačke u ravni. Naime, fontana i ($0 \leq i \leq n - 1$) se predstavlja tačkom $(x[i], y[i])$ gdje su $x[i]$ i $y[i]$ **parni brojevi**. Lokacije svih fontana su međusobno različite.

Arhitekta Timoti je unajmljen da dodatno uljepša park tako što će dodati neke **staze** i na svakoj od njih postaviti po jednu **klupu**.

Staza je **horizontalna** ili **vertikalna** linija dužine 2 čije su krajnje tačke dvije različite fontane. Staze se moraju napraviti tako da posjetilac može posjetiti svaku fontanu krećući se po stazama. Na početku nema staza u parku.

Na svakoj stazi potrebno je **postaviti tačno** jednu klupu. Svaka klupa se postavlja na lokaciju (a, b) gdje su a i b **neparni brojevi**. Lokacije klupa moraju biti **različite**. Klupa (a, b) može biti postavljena na stazu samo ako **obje** krajnje tačke puta pripadaju sljedećem skupu tačaka: $(a - 1, b - 1)$, $(a - 1, b + 1)$, $(a + 1, b - 1)$ i $(a + 1, b + 1)$. Na primjer, klupa $(3, 3)$ može biti postavljena na jednoj od sljedeće četiri staze: $(2, 2) - (2, 4)$, $(2, 4) - (4, 4)$, $(4, 4) - (4, 2)$, $(4, 2) - (2, 2)$.

Pomozite Timotiju da odredi da li je moguće konstruisati staze i postaviti klupe u skladu sa datim ograničenjima i ako jeste, prikažite rješenje. Ako postoji više rješenja, prikažite bilo koje od njih.

Detalji implementacije

Potrebno je implementirati sljedeću funkciju:

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

- x, y : nizovi dužine n . Za svako i ($0 \leq i \leq n - 1$), fontana i je u tački $(x[i], y[i])$, gdje su $x[i]$ i $y[i]$ parni brojevi.
- Ako je moguće konstruisati staze i postaviti klupe, potrebno je tačno jednom pozvati funkciju `build` (pogledajte ispod) koja prikazuje rješenje i zatim vraća `1`.
- Ako nije moguće konstruisati staze i postaviti klupe, funkcija vraća `0` ne pozivajući `build`.
- Ova se funkcija poziva tačno jednom.

Vaša implementacija može pozvati sledeću funkciju da bi prikazala staze i raspored klupa:

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

Označimo sa m broj konstruisanih staza.

- u, v : nizovi dužine m koji predstavljaju staze. Staze su označene brojevima od 0 do $m - 1$. Za svako j ($0 \leq j \leq m - 1$), staza j povezuje fontane $u[j]$ i $v[j]$. Svaka staza mora biti horizontalna ili vertikalna linija dužine 2 . Dvije različite staze mogu imati najviše jednu zajedničku tačku (naime, fontanu). Kada se konstruišu sve staze, biće moguće kretanje između bilo koje dvije fontane.
- a, b : nizovi dužine m koji predstavljaju klupe. Za svako j ($0 \leq j \leq m - 1$), lokacija klupe je $(a[j], b[j])$ i postavljena je na stazu j . Na jednoj lokaciji ne mogu biti dvije klupe. Na jednoj stazi ne mogu biti različite klupe.

Primjeri

Primjer 1

Razmotrimo sljedeći poziv funkcije:

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

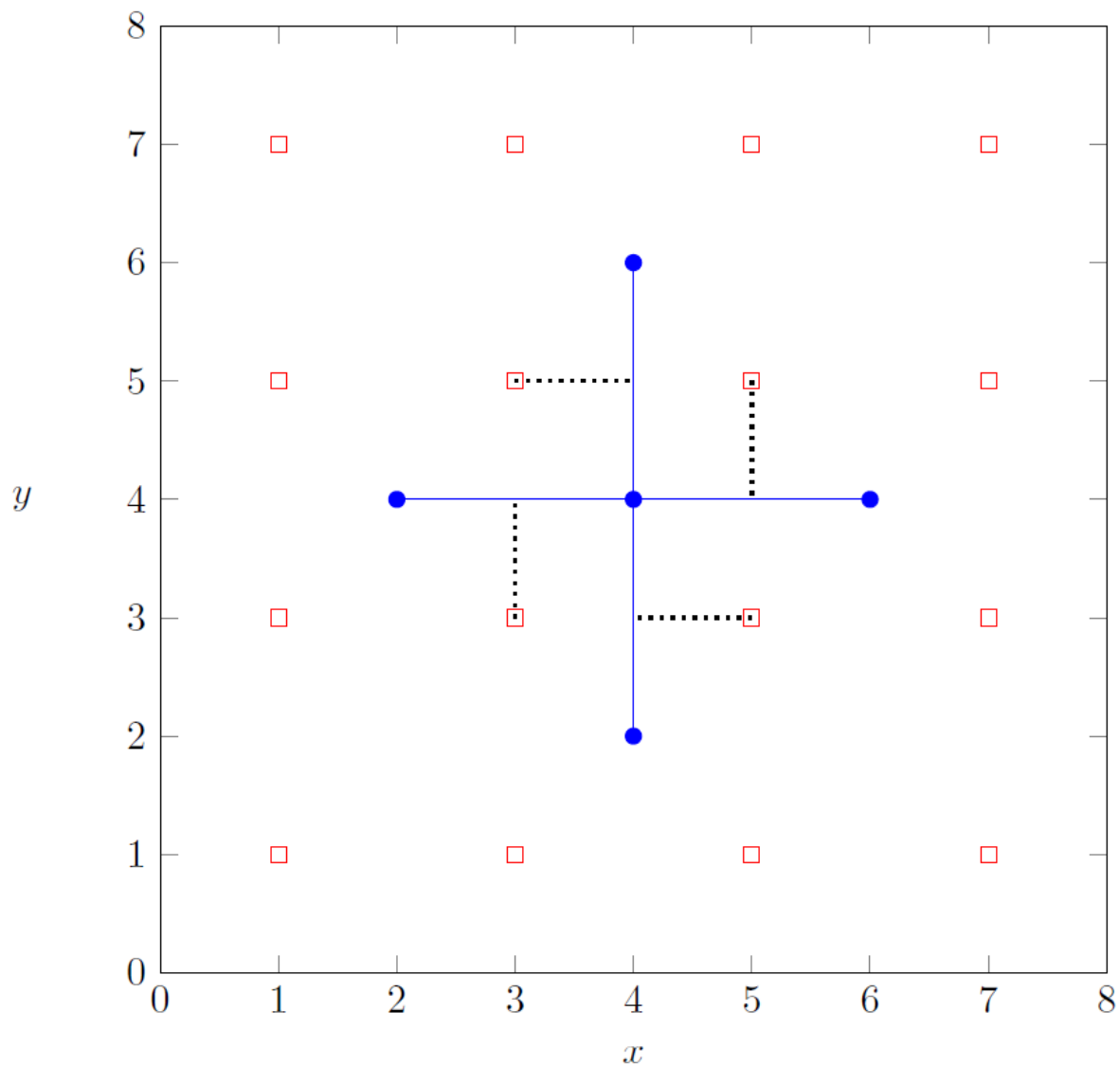
Postoji 5 fontana:

- fontana 0 je na lokaciji $(4, 4)$,
- fontana 1 je na lokaciji $(4, 6)$,
- fontana 2 je na lokaciji $(6, 4)$,
- fontana 3 je na lokaciji $(4, 2)$,
- fontana 4 je na lokaciji $(2, 4)$.

Moguće je konstruisati 4 staze, gdje svaka staza povezuje dvije fontane, i na svaku stazu postaviti odgovarajuće klupe:

Oznaka puta	Fontane koje povezuje staza	Lokacija postavljene klupe
0	0, 2	(5, 5)
1	0, 1	(3, 5)
2	3, 0	(5, 3)
3	4, 0	(3, 3)

Navedeno rješenje odgovara sljedećoj slici:



Da bi prikazali ovo rješenje, u funkciji `construct_roads` pozivamo `build` na sljedeći način:

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

Zatim vraćamo `1`.

Obratite pažnju da u ovom primjeru postoji više rješenja koja zadovoljavaju zahtjeve zadatka. Na primjer, ispravno je pozvati i `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` i zatim vratiti `1`.

Primjer 2

Razmotrimo sljedeći poziv funkcije:

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

Lokacija fontana `0` je `(2, 2)` a lokacija fontane `1` je `(4, 6)`. Kako ne postoji način da se konstruišu putevi koji zadovoljavaju sve zahtjeve, `construct_roads` vraća `0` i ne poziva `build`.

Ograničenja

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$ (za sve $0 \leq i \leq n - 1$)
- $x[i]$ i $y[i]$ su parni brojevi (za sve $0 \leq i \leq n - 1$).
- Dvije fontane ne mogu biti na istoj lokaciji.

Podzadaci

1. (5 bodova) $x[i] = 2$ (za sve $0 \leq i \leq n - 1$)
2. (10 bodova) $2 \leq x[i] \leq 4$ (za sve $0 \leq i \leq n - 1$)
3. (15 bodova) $2 \leq x[i] \leq 6$ (za sve $0 \leq i \leq n - 1$)
4. (20 bodova) Postoji najviše jedan način kontruisanja staza tako da se posjetilac parka može kretati između bilo koje dvije fontane.
5. (20 bodova) Ne postoje četiri fontane koje predstavljaju tjemena kvadrata 2×2 .
6. (30 bodova) Nema dodatnih ograničenja.

Program za ocjenjivanje (Sample Grader)

Format za učitavanje podataka je:

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

Format za štampanje je:

- red 1: vrijednost koju vraća `construct_roads`

Ako je vrijednost koju vraća `construct_roads` 1 i pozvana je `build(u, v, a, b)`, dodatno se štampa:

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