

## Fontanų parkas

Netoliese esančiame parke yra  $n$  fontanų, pažymėtų skaičiais nuo  $0$  iki  $n - 1$ . Fontanus žymime taškais dvimatėje plokštumoje. Kitaip tariant,  $i$ -asis ( $0 \leq i \leq n - 1$ ) fontanas yra taškas  $(x[i], y[i])$ , kur  $x[i]$  ir  $y[i]$  yra lyginiai skaičiai. Visų fontanų pozicijos yra skirtingos.

Architektas Timotis turi suprojektuoti, kaip nutiesti keletą takų bei kiekvienam takui pastatyti po vieną suoliuką. Takas yra **horizontali** arba **vertikali** 2 vienetų ilgio atkarpa, kurios galuose yra du skirtingi fontanai. Takai turėtų būti nutiesti taip, kad jais būtų galima nukeliauti tarp bet kurių dviejų fontanų. Pradiniu momentu parke takų nėra.

Kiekvienam takui turi būti pastatytas lygiai vienas suoliukas, kuris bus priskirtas tam takui (t.y., bus atsuktas į tą taką). Kiekvienas suoliukas privalo būti pastatytas kažkokiam taške  $(a, b)$ , kur  $a$  ir  $b$  yra nelyginiai sveikieji skaičiai. Visų suoliukų pozicijos privalo būti skirtingos. Taške  $(a, b)$  esantis suoliukas gali būti priskirtas takui tik tada, kai abu to tako galai yra tarp taškų  $(a - 1, b - 1)$ ,  $(a - 1, b + 1)$ ,  $(a + 1, b - 1)$  ir  $(a + 1, b + 1)$ . Pavyzdžiui, taške  $(3, 3)$  esantis suoliukas gali būti priskirtas takui, kuris atitinka vieną iš keturių atkarpų  $(2, 2) - (2, 4)$ ,  $(2, 4) - (4, 4)$ ,  $(4, 4) - (4, 2)$ ,  $(4, 2) - (2, 2)$ .

Padėkite Timočiui nustatyti, ar įmanoma nutiesti takus ir pastatyti suoliukus taip, kad būtų tenkinami reikalavimai, ir jeigu tai įmanoma, pateikite jam galimą statybų planą. Jei yra daugiau nei vienas reikalavimus tenkinantis sprendimas, galite pateikti bet kurį.

## Realizacija

Parašykite šią procedūrą:

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

- $x, y$ : du  $n$  ilgio masyvai. Kiekvienam  $i$  ( $0 \leq i \leq n - 1$ )  $i$ -asis fontanas yra taške  $(x[i], y[i])$ , kur  $x[i]$  ir  $y[i]$  yra lyginiai skaičiai.
- Jei statybos įmanomos, procedūra turėtų lygiai vieną kartą iškviesti procedūrą `build` (žiūrėti žemiau), kad pateiktų sprendimą, ir tada grąžinti `1`.
- Kitu atveju procedūra turėtų grąžinti `0` neiškviesdama procedūros `build`.
- Ši procedūra iškviečiama lygiai vieną kartą.

Jūsų procedūra (aprašyta aukščiau) gali kviešti šią procedūrą, kad pateiktų būdą nutiesti takus bei pastatyti suoliukus:

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

- Tiesiamų kelių skaičių pažymėkime  $m$ .
- $u, v$ : du  $m$  ilgio masyvai, nusakantys takus, kuriuos norite nutiesti. Šie takai sunumeruoti nuo 0 iki  $m - 1$ . Kiekvienam  $j$  ( $0 \leq j \leq m - 1$ )  $j$ -asis takas jungia  $u[j]$ -ąjį ir  $v[j]$ -ąjį fontanus. Kiekvienas takas privalo būti horizontali arba vertikali 2 vienetų ilgio atkarpa. Bet kurie du skirtingi takai gali turėti ne daugiau nei vieną bendrą tašką (fontaną). Kai takai bus pastatyti, jais turėtų būti galima nukeliauti tarp bet kurių dviejų fontanų.
- $a, b$ : du  $m$  ilgio masyvai, nusakantys suoliukus. Kiekvienam  $j$  ( $0 \leq j \leq m - 1$ ) pastatomas suoliukas taške  $(a[j], b[j])$ , kuris yra priskiriamas  $j$ -ajam takui. Jokie du skirtingi suoliukai negali būti toje pačioje pozicijoje.

## Pavyzdžiai

### Pavyzdys nr. 1

Panagrinėkime šį išskvietimą:

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

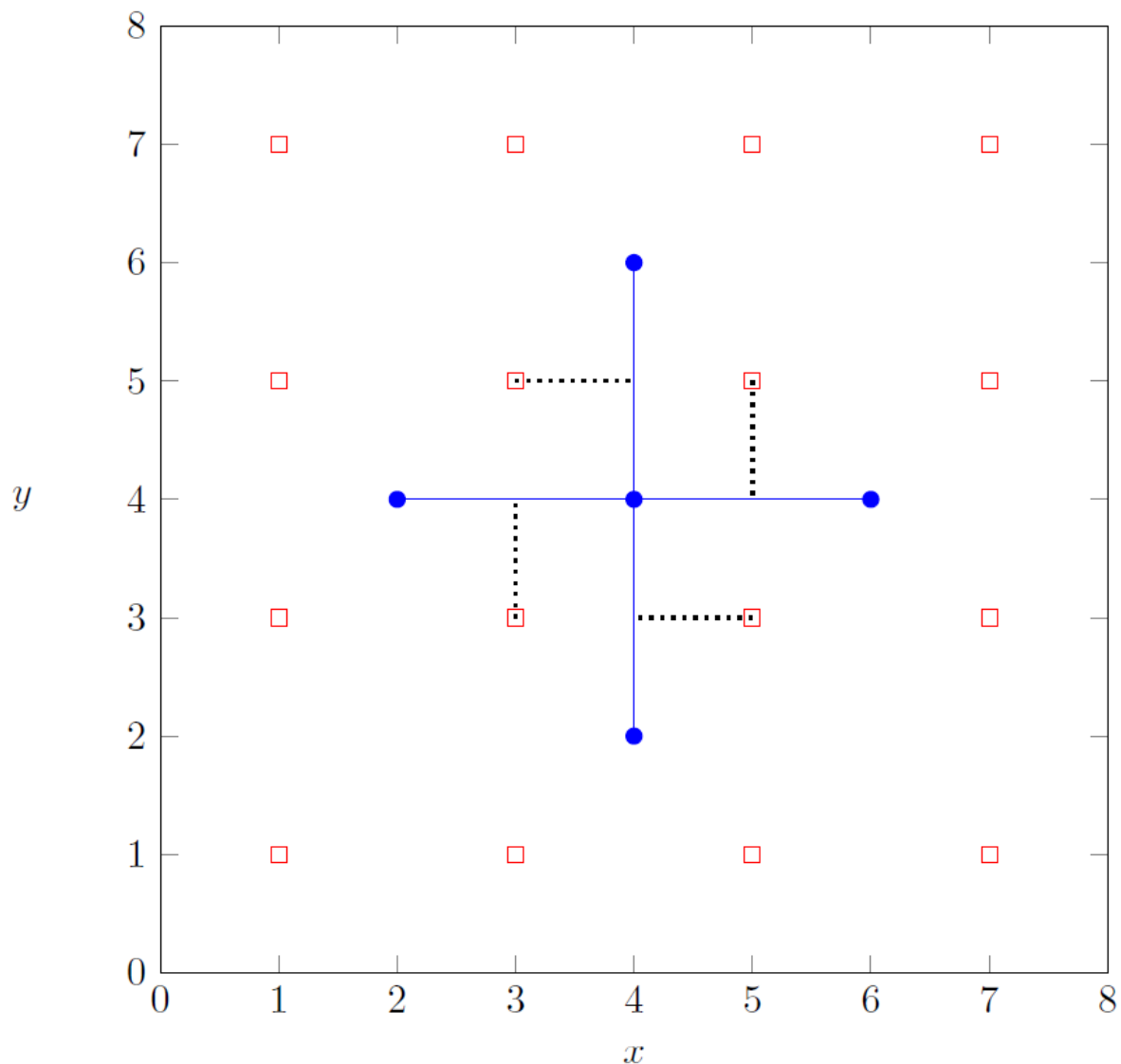
Tai reiškia, kad iš viso yra 5 fontanai:

- 0-inis fontanas yra taške  $(4, 4)$ ,
- 1-asis fontanas yra taške  $(4, 6)$ ,
- 2-asis fontanas yra taške  $(6, 4)$ ,
- 3-asis fontanas yra taške  $(4, 2)$ ,
- 4-asis fontanas yra taške  $(2, 4)$ .

Įmanoma nutiesti 4 takus taip, kad būtų galima nukeliauti tarp bet kurių dviejų fontanų, bei pastatyti atitinkamus suoliukus:

Tako numeris	Tako jungiamų fontanų numeriai	Priskirto suoliuko pozicija
0	0, 2	$(5, 5)$
1	0, 1	$(3, 5)$
2	3, 0	$(5, 3)$
3	4, 0	$(3, 3)$

Šis sprendimas atitinka šią diagramą:



Šį sprendimą pateikiame `construct_roads` procedūroje atlikdami tokį iškvietimą:

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

Procedūra tada turėtų grąžinti 1.

Atkreipkite dėmesį, kad šiuo atveju yra daugiau nei vienas reikalavimus atitinkantis sprendimas, ir bet kuris iš jų būtų teisingas. Pavyzdžiui, taip pat galima iškviešti `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` ir tada grąžinti 1.

## Pavyzdys nr. 2

Panagrinėkime šį iškvietimą:

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

0-inis fontanas yra pastatytas taške (2,2), o 1-asis fontanas taške (4,6). Kadangi nėra jokio būdo nutiesti takus, kurie tenkintų reikalavimus, procedūra `construct_roads` turėtų grąžinti 0

neiškviessdama procedūros `build`.

## Ribojimai

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$  (visiems  $0 \leq i \leq n - 1$ )
- $x[i]$  ir  $y[i]$  yra lyginiai sveikieji skaičiai (visiems  $0 \leq i \leq n - 1$ ).
- Jokie du fontanai nėra toje pačioje pozicijoje.

## Dalinės užduotys

1. (5 taškai)  $x[i] = 2$  (visiems  $0 \leq i \leq n - 1$ )
2. (10 taškų)  $2 \leq x[i] \leq 4$  (visiems  $0 \leq i \leq n - 1$ )
3. (15 taškų)  $2 \leq x[i] \leq 6$  (visiems  $0 \leq i \leq n - 1$ )
4. (20 taškų) Yra daugiausiai vienas būdas nutiesti takus, kad jais būtų galima nukeliauti tarp bet kurių dviejų fontanų.
5. (20 taškų) Jokie keturi fontanai nėra pastatyti to paties  $2 \times 2$  kvadrato kampuose.
6. (30 taškų) Jokių papildomų ribojimų.

## Pavyzdinė vertinimo programa

Pavyzdinė vertinimo programa skaito įvestį šiuo formatu:

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

Pavyzdinė vertinimo programa išveda atsakymą šiuo formatu:

- 1-oji eilutė: `construct_roads` gražinta vertė

Jei `construct_roads` gražinta vertė yra 1 ir `build(u, v, a, b)` buvo iškvieštas, vertinimo programa papildomai išveda šias eilutes:

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