

# Purskkaevupargid

Lähedalasuvas pargis asuvad  $n$  **purskkaevu**, nummerdatud  $0$  kuni  $n - 1$ . Modelleerime purskkaeve kui punkte kahemõõtmelisel tasandil. Täpsemini on purskkaev  $i$  ( $0 \leq i \leq n - 1$ ) punkt  $(x[i], y[i])$ , kus  $x[i]$  ja  $y[i]$  on **paaris täisarvud**. Kõikide purskkaevude asukohad on erinevad.

Arhitekt Timothy on palgatud rajama mõningaid **teid** ja üht **pink**i iga tee kohta. Tee on **horisontaalne** või **vertikaalne** sirgloik pikkusega  $2$ , mille otspunktideks on kaks eri purskkaevu. Teed tuleb ehitada nii, et igast purskkaevust oleks võimalik mööda teid liikuda iga teise purskkaevuni. Algselt ei ole pargis ühtegi teed.

Iga tee jaoks on vaja parki panna **täpselt** üks pink, mis on sellele teele **määratud** (st suunatud selle poole). Iga pink peab asuma mingis punktis  $(a, b)$ , kus  $a$  ja  $b$  on **paaritud täisarvud**. Kõikide pinkide asukohad peavad olema **erinevad**. Pink  $(a, b)$  saab olla määratud tee ainult siis, kui **mõlemad** tee otspunktid kuuluvad hulka  $(a - 1, b - 1)$ ,  $(a - 1, b + 1)$ ,  $(a + 1, b - 1)$  ja  $(a + 1, b + 1)$ . Näiteks võib pink punktis  $(3, 3)$  olla määratud ainult ühele neljast järgmisest teelõigust:  $(2, 2) - (2, 4)$ ,  $(2, 4) - (4, 4)$ ,  $(4, 4) - (4, 2)$ ,  $(4, 2) - (2, 2)$ .

Aita Timothy kindlaks teha, kas on võimalik ehitada teid ning asetada ja määrata pinke nii, et kõik ülaltoodud tingimused on rahuldatud. Kui on, siis paku talle välja üks sobiv lahendus. Kui sobivaid lahendusi on mitu, siis võid välja pakkuda ükskõik missuguse neist.

## Realisatsioon

Lahendusena tuleb realiseerida funktsioon:

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

- $x, y$ : kaks massiivi pikkusega  $n$ . Iga  $i$  ( $0 \leq i \leq n - 1$ ) jaoks on purskkaev punkt  $(x[i], y[i])$ , kus  $x[i]$  ja  $y[i]$  on paaris täisarvud.
- Kui lahendus leidub, siis peab see funktsioon kutsuma välja funktsiooni `build` täpselt ühe korra (vt all), et pakkuda välja lahendus, pärast mida peab funktsioon tagastama `1`.
- Vastasel juhul peaks funktsioon tagastama `0` ilma ühegi funktsiooni `build` väljakutseta.
- Seda funktsiooni kutsutakse välja täpselt üks kord.

Sinu realisatsioon võib kutsuda välja järgmist funktsiooni, et pakkuda välja sobiv lahendus teede ja pinkide paigutamiseks:

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

- Olgu  $m$  lahenduses olevate teede arv.

- $u, v$ : kaks massiivi pikkusega  $m$ , mis kirjeldavad ehitatavaid teid. Need teed on nummerdatud 0 kuni  $m - 1$ . Iga  $j$  ( $0 \leq j \leq m - 1$ ) jaoks ühendab tee  $j$  purskkaeve  $u[j]$  ja  $v[j]$ . Iga tee peab olema horisontaalne või vertikaalne sirglõik pikkusega 2. Kaks eri teed võivad omada ülimalt üht ühist punkti (purskkaev). Kui teed on ehitatud, siis peab olema võimalik teid mööda reisida ükskõik missguse kahe purskkaevu vahel.
- $a, b$ : kaks massiivi pikkusega  $m$ , mis kirjeldavad pinke. Iga  $j$  ( $0 \leq j \leq m - 1$ ) jaoks asetatakse pink punkti  $(a[j], b[j])$  ja määratakse teele  $j$ . Kaks eri pinki ei tohi asuda samas kohas.

## Näited

### Näide 1

Vaatame järgmist väljakutset:

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

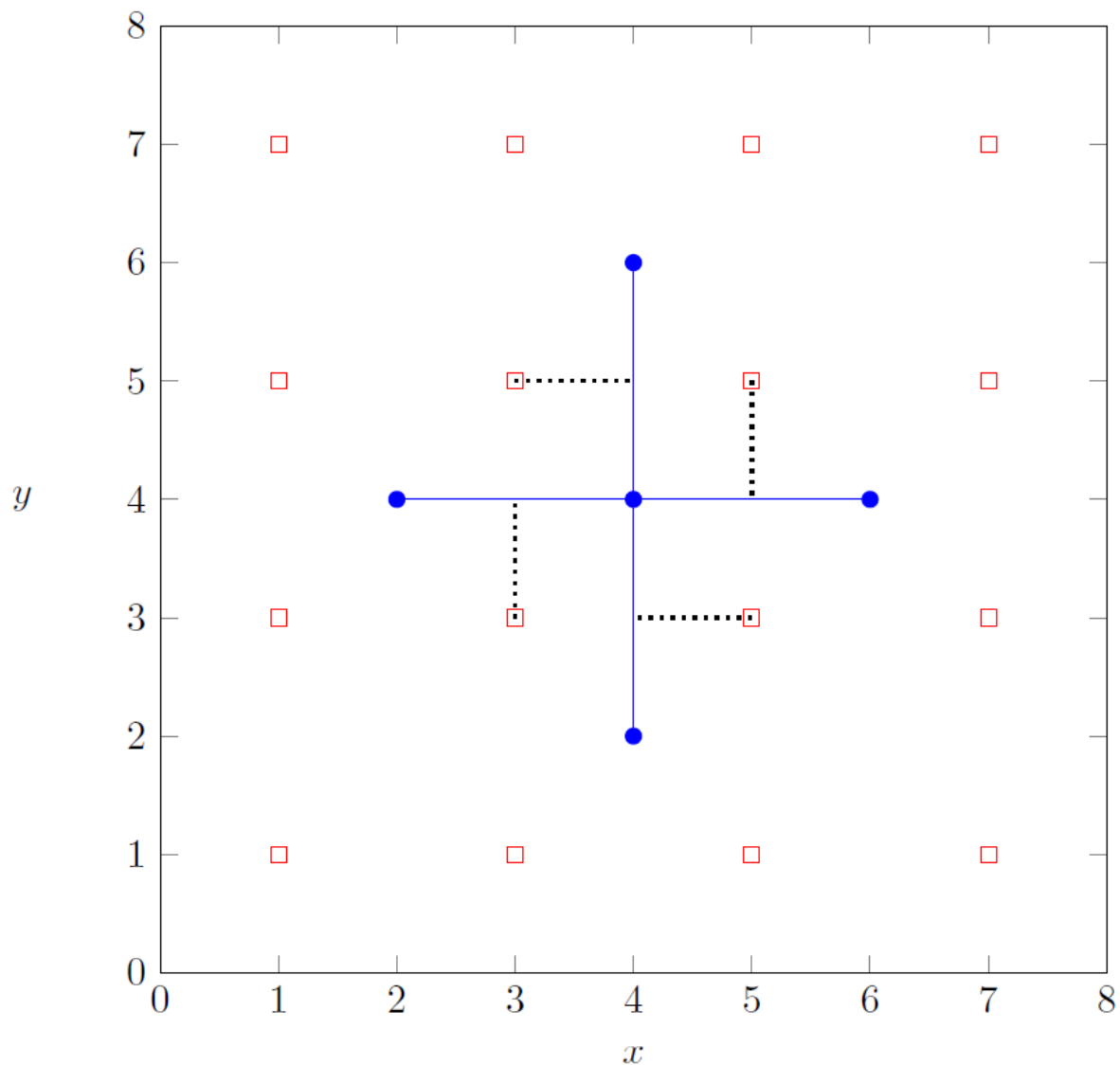
See tähendab, et purskkaeve on 5:

- purskkaev 0 asub punktis  $(4, 4)$ ,
- purskkaev 1 asub punktis  $(4, 6)$ ,
- purskkaev 2 asub punktis  $(6, 4)$ ,
- purskkaev 3 asub punktis  $(4, 2)$ ,
- purskkaev 4 asub punktis  $(2, 4)$ .

Võimalik on ehitada järgmised 4 teed, kus iga tee ühendab kaht purskkaevu, ja asetada vastavad pingid:

Tee number	Purskkaevud, mida tee ühendab	Määratud pingi asukoht
0	0, 2	$(5, 5)$
1	0, 1	$(3, 5)$
2	3, 0	$(5, 3)$
3	4, 0	$(3, 3)$

See lahendus vastab järgmisele skeemile:



Lahenduse väljapakkumiseks peaks `construct_roads` tegema järgmise väljakutse:

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

Pärast seda peaks ta tagastama 1.

Sellele näitele leidub palju lahendusi, mis kõik rahuldavad ülesande tingimusi ja mida kõiki loetakse õigeks. Näiteks on õige väja kutsuda `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` ja siis tagastada 1.

## Näide 2

Vaatame järgmist väljakutset:

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

Purskkaev 0 asub punktis (2,2) ja purskkaev 1 punktis (4,6). Kuna ei ole olemas viisi ehitada teid tingimustele vastaval viisil, siis peab `construct_roads` tagastama 0 ilma funktsiooni `build`

välja kutsumata.

## Piirangud

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$  (kõigi  $0 \leq i \leq n - 1$  jaoks)
- $x[i]$  ja  $y[i]$  on paarisarvulised täisarvud (kõigi  $0 \leq i \leq n - 1$  jaoks)
- Ühelgi kahel purskkaevul ei ole sama asukoht

## Alamülesanded

1. (5 punkti)  $x[i] = 2$  (kõigi  $0 \leq i \leq n - 1$  jaoks)
2. (10 punkti)  $2 \leq x[i] \leq 4$  (kõigi  $0 \leq i \leq n - 1$  jaoks)
3. (15 punkti)  $2 \leq x[i] \leq 6$  (kõigi  $0 \leq i \leq n - 1$  jaoks)
4. (20 punkti) Teede paigutamiseks nõnda, et iga kahe purskkaevu vahel on võimalik teid mööda reisida, on ülimalt üks võimalus.
5. (20 punkti) Ei leidu nelja purskkaevu, mis asuvad  $2 \times 2$  ruudu nurkades.
6. (30 punkti) Lisapiirangud puuduvad.

## Näidishindaja

Näidishindaja loeb sisendit järgmises vormingus:

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

Näidishindaja väljastab tulemuse järgmises vormingus:

- rida 1: funktsiooni `construct_roads` tagastatud väärtus

Kui funktsiooni `construct_roads` tagastatud väärtus on 1 ja `build(u, v, a, b)` on välja kutsutud, siis väljastab näidishindaja lisaks ka

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