

Strūklaku parks

Tuvējā parkā atrodas n **strūklakas**, kas sanumurētas no 0 līdz $n - 1$. Modelēsim strūklakas kā punktus divdimensiju plaknē. Tas ir, strūklaka i ($0 \leq i \leq n - 1$) ir punkts $(x[i], y[i])$, kur $x[i]$ un $y[i]$ ir **pāra skaitļi**. Visu strūklaku koordinātas ir atšķirīgas.

Arhitekts Timotijs ir nolīgts, lai izstrādātu plānu vairāku **ceļu** būvēšanai un viena **sola** izvietojšanai pie katra ceļa. Ceļš ir **horizontāls** vai **vertikāls** nogrieznis garumā 2 , kura galapunkti ir divas dažādas strūklakas. Ceļiem ir jābūt uzbūvētiem tādā veidā, lai, ejot pa ceļiem, var pārvietoties starp jebkurām divām strūklakām. Sākumā parkā ceļu nav.

Pie katra no ceļiem parkā ir jāizvieto **tieši** viens solis, un šis solis ir **jāpiešķir** šim ceļam (t.i., jāpavērš pret šo ceļu). Katrs solis ir jāizvieto kādā punktā (a, b) , tā, lai a un b būtu **nepāra skaitļi**. Visu solu atrašanās vietām ir jābūt atšķirīgām. Solis punktā (a, b) var būt piešķirts ceļam tikai tad, ja **abi** ceļa galapunkti ir starp $(a - 1, b - 1)$, $(a - 1, b + 1)$, $(a + 1, b - 1)$ un $(a + 1, b + 1)$. Piemēram, solis punktā $(3, 3)$ var būt piešķirts tikai ceļam, kas ir viens no nogriežņiem $(2, 2) - (2, 4)$, $(2, 4) - (4, 4)$, $(4, 4) - (4, 2)$, $(4, 2) - (2, 2)$.

Palīdziet Timotijam noteikt, vai ir iespējams uzbūvēt ceļus, kā arī izvietot un piešķirt solus, ievērojot visas augstākminētās prasības, un, ja ir, tad uzrādiet viņam iespējamo risinājumu. Ja ir vairāki iespējamie risinājumi, kas apmierina visas prasības, jūs varat paziņot jebkuru no tiem.

Realizācijas detaļas

Jums ir jārealizē šāda procedūra:

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

- x, y : divi masīvi garumā n . Katram i ($0 \leq i \leq n - 1$) strūklaka i ir punkts $(x[i], y[i])$, kur $x[i]$ un $y[i]$ ir pāra skaitļi.
- Ja ceļus uzbūvēt ir iespējams, šai procedūrai ir jāizpilda tieši viens procedūras `build` izsaukums (skat. zemāk), lai paziņotu risinājumu, un pēc tam jāatgriež `1`.
- Citādi procedūrai ir jāatgriež `0`, neveicot nevienu procedūras `build` izsaukumu.
- Šī procedūra tiek izsaukta tieši vienreiz.

Jūsu realizācija var izsaukt šādu procedūru, lai paziņotu iespējamo ceļu būvēšanas un solu izvietojšanas plānu:

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

- u, v : divi masīvi garumā m , kas atbilst būvējamajiem ceļiem. Ceļi ir numurēti no 0 līdz $m - 1$. Katram j ($0 \leq j \leq m - 1$) ceļš j savieno strūklakas $u[j]$ un $v[j]$. Katra ceļam ir jābūt horizontālam vai vertikālam nogrieznim garumā 2 . Jebkuriem diviem atšķirīgiem ceļiem ir jāšķēļas ne vairāk kā vienā punktā (strūklakā). Kad ceļi ir uzbūvēti, ir jāvar pa ceļiem pārvietoties starp jebkurām divām strūklakām.
- a, b : divi masīvi garumā m , kas apzīmē solus. Katram j ($0 \leq j \leq m - 1$) solis ir novietots punktā $(a[j], b[j])$ un piešķirts ceļam j . Nekādi divi atšķirīgi soli nevar atrasties vienā punktā. Dažādi soli nevar būt piešķirti vienam ceļam.

Piemēri

1. piemērs

Apskatīsim šādu izsaukumu:

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

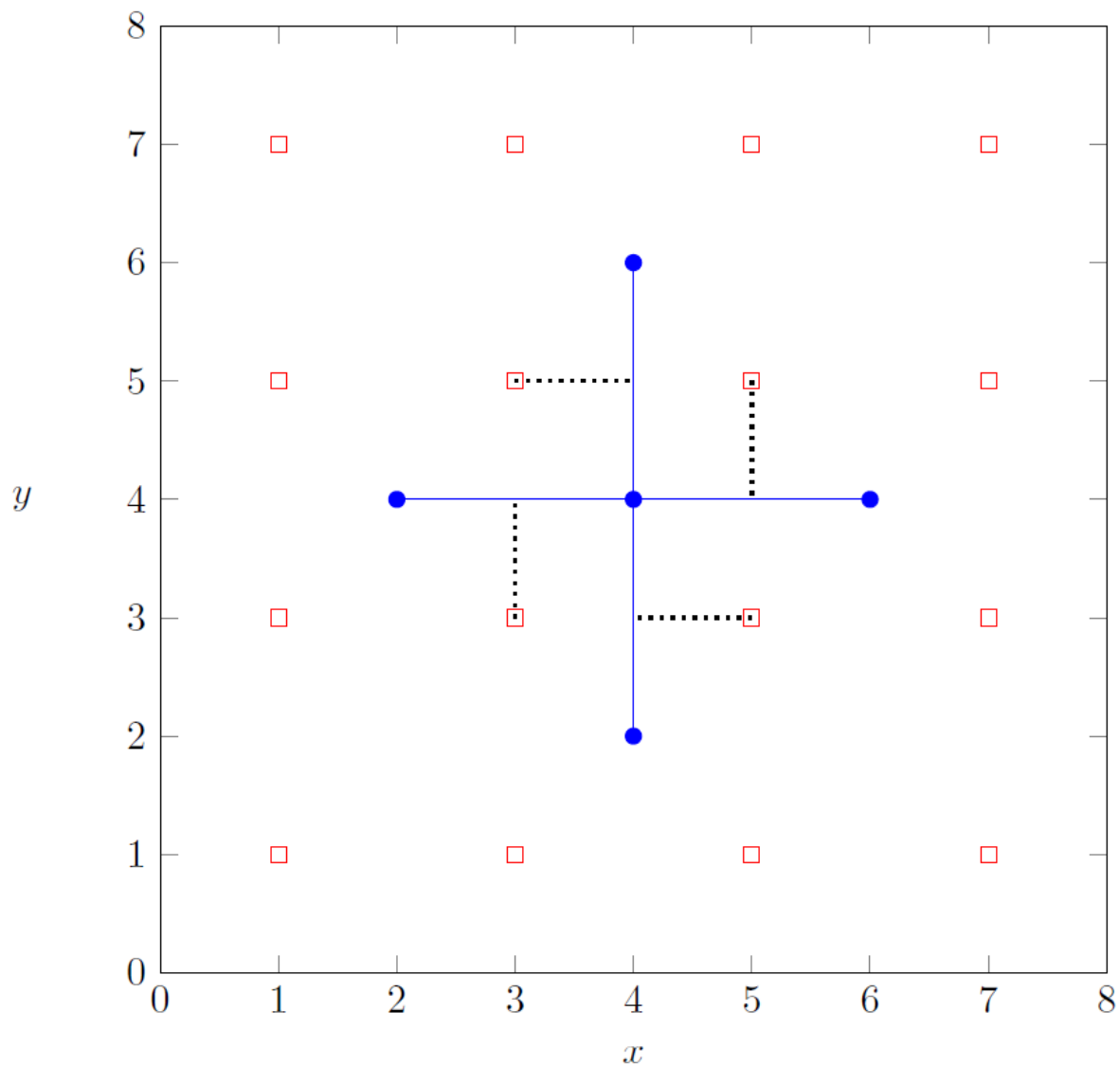
Tas nozīmē, ka ir 5 strūklakas:

- strūklaka 0 ir izvietota punktā $(4, 4)$,
- strūklaka 1 ir izvietota punktā $(4, 6)$,
- strūklaka 2 ir izvietota punktā $(6, 4)$,
- strūklaka 3 ir izvietota punktā $(4, 2)$,
- strūklaka 4 ir izvietota punktā $(2, 4)$.

Ir iespējams uzbūvēt tādus 4 ceļus, kur katrs ceļš savieno divas strūklakas, un izvietot atiecīgos solus:

Ceļa numurs	Strūklaku numuri, kuras savieno ceļš	Piešķirtā sola atrašanās vieta
0	0, 2	(5, 5)
1	0, 1	(3, 5)
2	3, 0	(5, 3)
3	4, 0	(3, 3)

Šis risinājums atbilst šādam zīmējumam:



Lai paziņotu šo risinājumu, procedūrai `construct_roads` ir jāveic šāds izsaukums:

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

Tad tai ir jāatgriež `1`.

Ievērojiet, ka šajā gadījumā ir vairāki risinājumi, kas apmierina prasības, un tie visi tiktu uzskatīti par pareiziem. Piemēram, iespējams arī izsaukt `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` un tad atgriezt `1`.

2. piemērs

Apskatīsim šādu izsaukumu:

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

Strūklaka `0` ir izvietota punktā $(2, 2)$, un strūklaka `1` ir izvietota punktā $(4, 6)$. Tā kā nevar uzbūvēt ceļus tā, lai apmierinātu visas prasības, `construct_roads` ir jāatgriež `0`, nevienu reizi neizsaucot

build.

Ierobežojumi

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$ (visiem $0 \leq i \leq n - 1$)
- $x[i]$ un $y[i]$ ir pāra skaitļi (visiem $0 \leq i \leq n - 1$).
- Nekādas divas strūklakas neatrodas vienā punktā.

Apakšuzdevumi

1. (5 punkti) $x[i] = 2$ (visiem $0 \leq i \leq n - 1$)
2. (10 punkti) $2 \leq x[i] \leq 4$ (visiem $0 \leq i \leq n - 1$)
3. (15 punkti) $2 \leq x[i] \leq 6$ (visiem $0 \leq i \leq n - 1$)
4. (20 punkti) Ir ne vairāk kā viens veids uzbūvēt ceļus tā, lai starp jebkurām divām strūklakām var pārvietoties, ejot pa ceļiem.
5. (20 punkti) Nav tādu četru strūklaku, kas atrastos 2×2 kvadrāta stūros.
6. (30 punkti) Bez papildu ierobežojumiem.

Paraugvērtētājs

Paraugvērtētājs lasa ievadu šādā formātā:

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

Paraugvērtētāja izvads ir šādā formātā:

- 1. rinda: procedūras `construct_roads` atgrieztā vērtība

Ja procedūras `construct_roads` atgrieztā vērtība ir 1 un procedūra `build(u, v, a, b)` tika izsaukta, tad vērtētājs papildus izvada:

- 2. rinda: m
- $(3 + i)$ -tā rinda: ($0 \leq i \leq m - 1$): $u[i] \ v[i] \ a[i] \ b[i]$