

## Fəvvarələr parkı

Yaxındakı parkda 0'dan  $n - 1$ 'ə qədər nömrələnmiş  $n$  ədəd **fəvvarə** var. Bu fəvvarələri 2 ölçülü müstəvidə nöqtələr kimi qeyd edirik. Yəni,  $i$  ( $0 \leq i \leq n - 1$ ) fəvvarəsi  $(x[i], y[i])$  nöqtəsidir,  $x[i]$  və  $y[i]$  **cüt** ədədlərdir. Bütün fəvvarələrin koordinatları müxtəlifdir.

Memar Timothy bəzi **yollar** inşa etməli və hər yol üçün bir **skamya** yerləşdirməlidir. Yol, uc nöqtələri müxtəlif fəvvarələr olan 2 uzunluqlu **horizontal** və ya **vertikal** parçadır. Yollar elə inşa olunmalıdır ki, bu yollardan istifadə edərək istənilən iki fəvvarə arasında səyahət etmək mümkün olsun. Başlanğıcda parkda yol yoxdur.

Hər yol üçün **yalnız** bir skamya yerləşdirmək lazımdır və bu skamya həmin yola **təyin** edilməlidir. Skamyalar, koordinatları **tək ədədlər** olan nöqtələrdə yerləşdirilə bilər. Bütün skamyaların koordinatları **müxtəlif** olmalıdır. Əgər yolun **hər iki** ucu  $(a - 1, b - 1)$ ,  $(a - 1, b + 1)$ ,  $(a + 1, b - 1)$ ,  $(a + 1, b + 1)$  nöqtələri arasından seçilibsə, onda  $(a, b)$  nöqtəsinə skamya yerləşdirmək və həmin yola təyin etmək olar. Məsələn,  $(3, 3)$  nöqtəsindəki skamyanı o yola təyin edə bilərsiniz ki, həmin yolun uc nöqtələri  $(2, 2) - (2, 4)$ ,  $(2, 4) - (4, 4)$ ,  $(4, 4) - (4, 2)$ ,  $(4, 2) - (2, 2)$  cütlərində biri olsun.

Timothy'ə yuxarıdakı şərtləri ödəyən yollar və skamyalar qurmağın mümkün olub olmamasını tapmaqda yardım et və əgər mümkündürsə, həll yolu göstər. Əgər bir neçə mümkün cavab varsa, hər hansı birini seçə bilərsiniz.

## İmplementasiya Detalları

Aşağıdakı prosedurları yerinə yetirməlisiniz:

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

- $x, y$ :  $n$  uzunluqlu iki massiv. Hər  $i$  ( $0 \leq i \leq n - 1$ ) üçün,  $i$ 'ci fəvvarə  $(x[i], y[i])$  nöqtəsindədir.  $x[i]$  və  $y[i]$  cüt ədədlərdir.
- Əgər mümkün konstruksiya varsa, bu prosedur `build` (daha ətraflı aşağıda) prosedurunu yalnızca bir dəfə çağıraraq həll yolunu bildirməlidir. Daha sonra 1 qaytarmalıdır.
- Əks halda bu prosedur `build` prosedurunu çağırmadan 1 qaytarmalıdır.
- Bu prosedur yalnızca 1 dəfə çağırılır.

Sənin kodun həll təqdim etmək üçün növbəti proseduru çağıra bilər:

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

- $m$  konstruksiyadakı bütün yolların sayını göstərsin.

- $u, v$ :  $m$  uzunluqlu iki massiv, inşa olunacaq yolları təmsil edirlər. Bu yollar  $0$ -dan  $m - 1$ -ə qədər işarələniblər. Hər  $j$  ( $0 \leq j \leq m - 1$ ) üçün,  $j$ -ci yol  $u[j]$  və  $v[j]$  fəvvarələri arasında əlaqə yaradır. Hər bir yol  $2$  uzunluqlu horizontal və ya vertikal parça olmalıdır. İstənilən iki yol arasında yalnızca bir ortaq nöqtə ola bilər, və bu nöqtə fəvvarə ola bilər. Yollar quraşdırıldıqdan sonra istənilən iki fəvvarə arasında səyahət etmək mümkün olmalıdır.
- $a, b$ :  $m$  uzunluqlu iki massiv, skamyaları təmsil edirlər. Hər  $j$  ( $0 \leq j \leq m - 1$ ) üçün,  $(a[j], b[j])$  nöqtəsinə skamya qoyulub və  $j$ -ci yola təyin olunub. İki müxtəlif skamya eyni nöqtədə yerləşə bilməz.

## Nümunələr

### Nümunə 1

Aşağıdakı çağırışı baxaq:

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

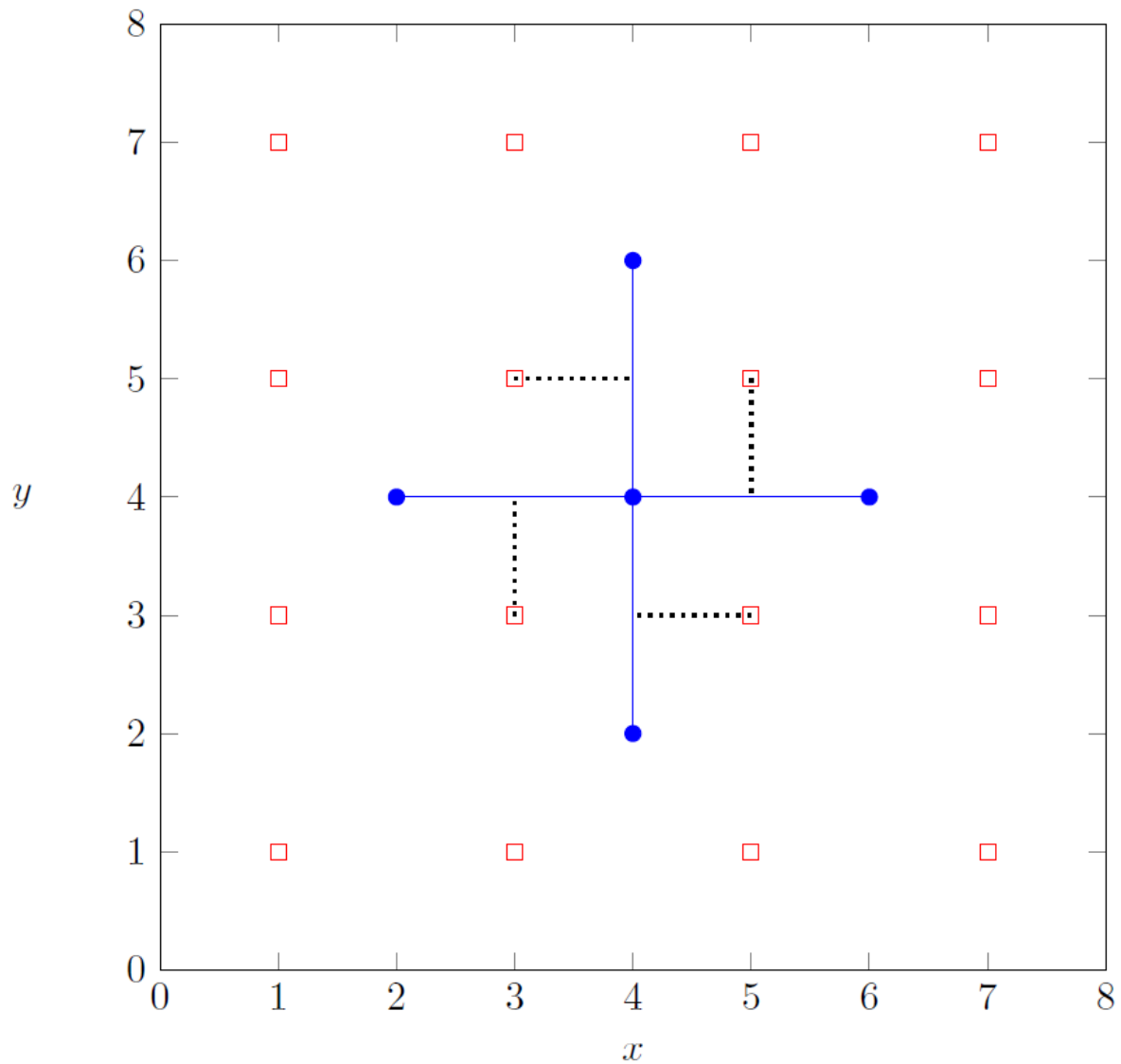
Bu o deməkdir ki, 5 fəvvarə var:

- fəvvarə 0, (4, 4) nöqtəsindədir,
- fəvvarə 1, (4, 6) nöqtəsindədir,
- fəvvarə 2, (6, 4) nöqtəsindədir,
- fəvvarə 3, (4, 2) nöqtəsindədir,
- fəvvarə 4, (2, 4) nöqtəsindədir.

Növbəti 4 yolu inşa edib, hər yola bir skamya təyin etmək olar:

Road label	Labels of the fountains the road connects	Location of the assigned bench
0	0, 2	(5, 5)
1	0, 1	(3, 5)
2	3, 0	(5, 3)
3	4, 0	(3, 3)

Bu həll aşağıdakı diaqrama aiddir:



Bu həlli təqdim etmək üçün, `construct_roads` növbəti çağırışı etməlidir:

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

Sonra, prosedur 1 qaytarmalıdır.

Qeyd etmək lazımdır ki, şərtləri ödəyən bir neçə həll yolu var, və hər biri düzgün cavab olaraq qəbul edilir. Məsələn, `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` çağırılıb, sonra 1 qaytarmaq da olardı.

## Nümunə 2

Növbəti çağırışa baxaq:

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

0'cı fəvvarə (2, 2) nöqtəsində, 1'ci fəvvarə isə (4, 6) nöqtəsindədir. İnşa edilə biləcək yol olmadığından, şərtləri ödəyən konstruksiya da yoxdur. Beləcə, `construct_roads`, `build`

prosesduruunu çağırmadan 0 qaytarmalıdır.

## Limitlər

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$  ( $0 \leq i \leq n - 1$ )
- $x[i]$  və  $y[i]$  cüt ədədlərdir ( $0 \leq i \leq n - 1$ ).
- Hər bir fəvvarə müxtəlif korrdinalardadır.

## Alt tapşırıqlar

1. (5 xal)  $x[i] = 2$  ( $0 \leq i \leq n - 1$ )
2. (10 xal)  $2 \leq x[i] \leq 4$  ( $0 \leq i \leq n - 1$ )
3. (15 xal)  $2 \leq x[i] \leq 6$  ( $0 \leq i \leq n - 1$ )
4. (20 xal) Bütün fəvvarələr arasında səyahət etməyin mümkün olması üçün yolların inşa edilməsinin yalnızca bir yolu var.
5. (20 xal) Heç bir 4 fəvvarə  $2 \times 2$  ölçülü kvadratın təpə nöqtələrində yerləşmir. There do not exist four fountains that form the corners of a square.
6. (30 xal) Əlavə məhdudiyyətlər yoxdur.

## Nümunə Grader (Qiymətləndirici)

Nümunə grader girişi bu formatda oxuyur

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

Nümunə grader bu formatda çıxışa verir:

- sətir 1: the return value of `construct_roads`

Əgər `construct_roads` 1 qaytararsa və `build(u, v, a, b)` çağırılsa, grader əlavə olaraq bunları çıxışa verəcək:

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