შადრევნებიანი პარკები

ახლოს მდებარე პარკში n **შადრევანია**, გადანომრილი 0-დან ($n-1$)-მდე. შადრევნები წარმოვიდგინოთ, როგორც ორგანზომილებიანი სიბრტყის წერტილები. კერძოდ, შადრევანი i ($0 \leq i \leq n-1$) არის $(x[i], y[i])$ წერტილი, სადაც $x[i]$ და $y[i]$ **ლუწი მთელი რიცხვებია**. შადრევნების მდებარეობები განსხვავებულია.

არქიტექტორი ტიმოთის მოვალეობაა შეადგინოს გეგმა **გზების** დასაგებად და თითოეული გზისთვის ერთი **სკამის** დასადგმელად. გზა არის **ჰორიზონტალური** ან **ვერტიკალური** მონაკვეთი სიგრძით 2, რომლის ბოლოებიც ორი განსხვავებული შადრევანია. გზები ისე უნდა დაიგოს, რომ შესაძლებელი იყოს მათი საშუალებით ნებისმიერ ორ შადრევანს შორის გადაადგილება. თავიდან პარკში არცერთი გზა არაა.

თითოეული გზისთვის პარკში უნდა დაიდგას **ზუსტად** ერთი სკამი და **დაენიშნოს** (ე.ი. უყურებდეს) ამ გზას. თითოეული სკამი უნდა დაიდგას რაიმე (a, b) წერტილში, სადაც a და b **კენტი მთელი რიცხვებია**. სკამების მდებარეობები უნდა იყოს **განსხვავებული**. სკამი (a, b) შეიძლება დაენიშნოს გზას მხოლოდ მაშინ, როდესაც გზის **ორივე** ბოლო არის $(a-1, b-1)$, $(a-1, b+1)$, $(a+1, b-1)$ და $(a+1, b+1)$-ს შორის. მაგალითად, სკამი $(3, 3)$ შეიძლება დაენიშნოს მხოლოს გზას, რომელიც შემდეგი ოთხი მონაკვეთიდან რომელიმეა: $(2, 2)$ – $(2, 4)$, $(2, 4)$ – $(4, 4)$, $(4, 4)$ – $(4, 2)$, $(4, 2)$ – $(2, 2)$.

დაეხმარეთ ტიმოთის დაადგინოს შესაძლებელია თუ არა გზების დაგება, სკამების დადგმა და მათი დანიშვნა ისე, რომ ყველა პირობა დაკმაყოფილდეს და თუ ეს შესაძლებელია, მოახსენეთ მას შესაძლო ამონახსნი. თუ პირობებს ერთზე მეტი ამონახსნი აკმაყოფილებს, თქვენ შეგიძლიათ შეარჩიოთ ნებისმიერი მათგანი.

იმპლემენტაციის დეტალები

თქვენ უნდა მოახდინოთ შემდეგი პროცედურის იმპლემენტაცია:

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

- x, y: ორი მასივი ზომით n. ყოველი i ($0 \leq i \leq n-1$)-თვის შადრევანი i არის წერტილი $(x[i], y[i])$, სადაც $x[i]$ და $y[i]$ ლუწი მთელი რიცხვებია.
- თუ აგება შესაძლებელია, პროცედურამ ზუსტად ერთხელ უნდა გამოიძახოს `build` (იხილეთ ქვემოთ) ამონახსნის მოსახსენებლად, რის შემდეგაც უნდა დააბრუნოს 1.
- წინააღმდეგ შემთხვევაში პროცედურამ 0 უნდა დააბრუნოს `build`-ის გამოუძახებლად.
- პროცედურა იქნება გამოძახებული ზუსტად ერთხელ.

თქვენმა იმპლემენტაციამ უნდა გამოიძახოს შემდეგი პროცედურა გზების დაგებისა და სკამების დადგმის შესაძლო ვარიანტის მოსახსენებლად:

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

- დავუშვათ, m არის დაგებული გზების რაოდენობა.
- u, v: ორი მასივი სიგრძით m, რომლებიც წარმოადგენენ დასაგებ გზებს. ეს გზები დანომრილია 0-დან ($m-1$)-მდე. ყოველი j ($0 \leq j \leq m-1$)-თვის გზა j აერთებს შადრევნებს ნომრებით $u[j]$ და $v[j]$. გზა უნდა იყოს ჰორიზონტალური ან ვერტიკალური მონაკვეთი სიგრძით 2. ორ სხვადასხვა გზას შეიძლება ჰქონდეს არაუმეტეს ერთი საერთო წერტილი (შადრევანი). გზების დაგების შემდეგ ყოველ ორ შადრევანს შორის გადაადგილება უნდა იყოს შესაძლებელი გზების დახმარებით.
- a, b: ორი მასივი ზომით m, რომლებიც წარმოადგენენ სკამებს. ყოველი j ($0 \leq j \leq m-1$)-თვის სკამი დაიდგმება წერტილში $(a[j], b[j])$ და დაენიშნება გზას ნომრით j. ორი განსხვავებული სკამი არ შეიძლება დაიდგას ერთი და იმავე ადგილას. სხვადასხვა სკამები არ შეიძლება დაენიშნოს ერთი და იმავე გზას.

მაგალითები

მაგალითი 1

განვიხილოთ შემდეგი გამოძახება:

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

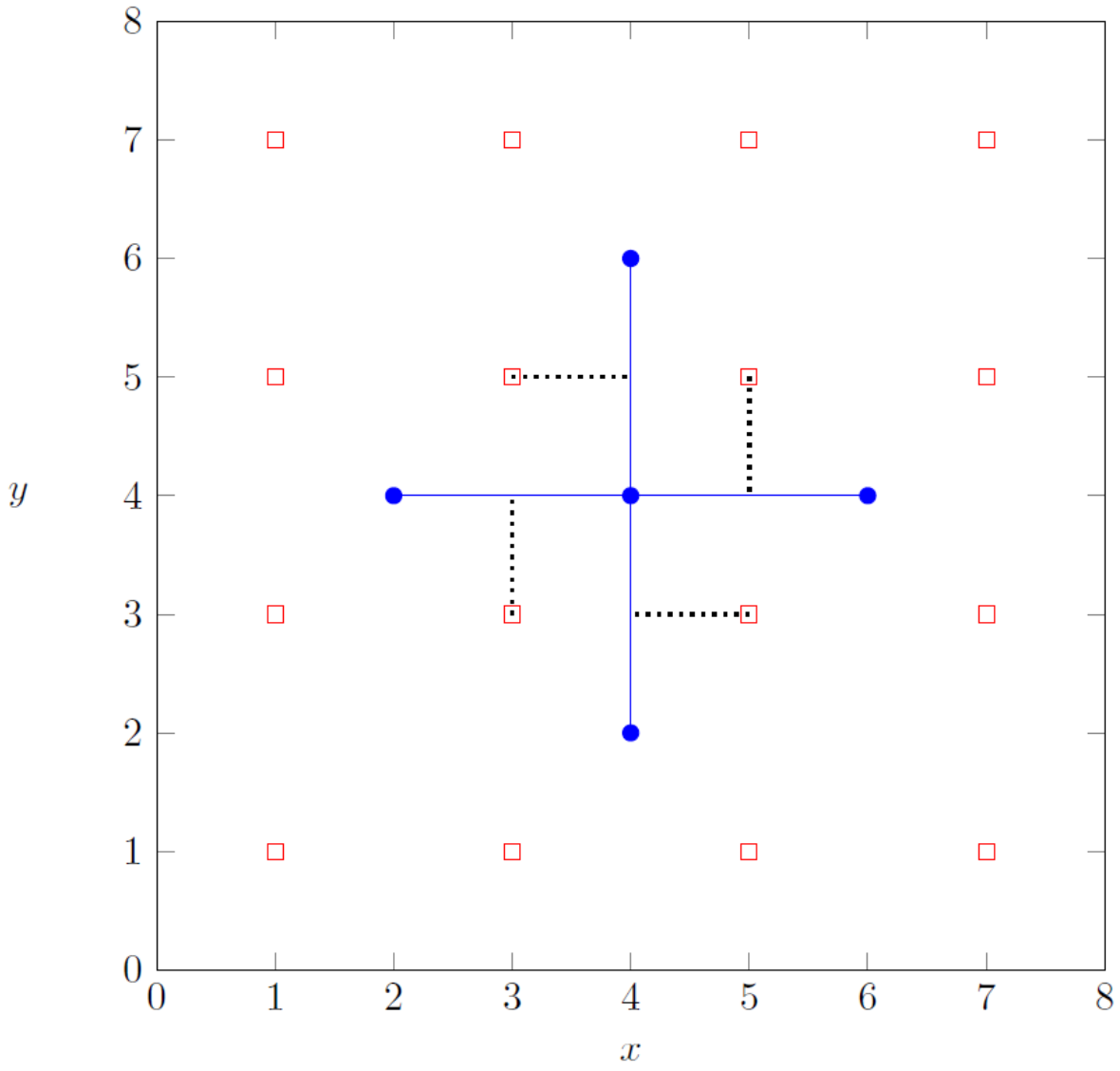
ეს ნიშნავს რომ არის 5 შადრევანი:

- შადრევანი 0 არის წერტილში $(4, 4)$,
- შადრევანი 1 არის წერტილში $(4, 6)$,
- შადრევანი 2 არის წერტილში $(6, 4)$,
- შადრევანი 3 არის წერტილში $(4, 2)$,
- შადრევანი 4 არის წერტილში $(2, 4)$.

შესაძლებელია 4 გზის დაგება, სადაც თითოეული აერთებს ორ შადრევანს და შემდეგი სკამების დადგმა:

გზის ნომერი	შეერთებული შადრევნების ნომრები	დანიშნული სკამების მდებარეობები
0	$0, 2$	$(5, 5)$
1	$0, 1$	$(3, 5)$
2	$3, 0$	$(5, 3)$
3	$4, 0$	$(3, 3)$

ამონახსნი შეესაბამება შემდეგ დიაგრამას:



ამ ამონახსნის მოსახენებლად `construct_roads` ახდებს შემდეგ გამოძახებას:

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

შემდეგ მან უნდა დააბრუნოს 1.

შევნიშნოთ, რომ ამ შემთხვევაში ამოცანის პირობას რამდენიმე ამონახსნი აკმაყოფილებს და ყველა მათგანი სწორად ჩაითვლება. მაგალითად, ასევე დასაშვებია გამოვიძახოთ `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` და შემდეგ დავაბრუნოთ 1.

მაგალითი 2

განვიხილოთ შემდეგი გამოძახება:

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

შადრევანი 0 არის წერტილში $(2, 2)$ და შადრევანი 1 არის წერტილში $(4, 6)$. რადგან შეუძლებელია გზების აგება ისე, რომ ყველა პირობა დაკმაყოფილდეს, `construct_roads` უნდა

აბრუნებდეს 0-ს `build`-ის გამოუძახებლად.

შეზღუდვები

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$ (ყოველი $0 \leq i \leq n-1$-თვის)
- $x[i]$ და $y[i]$ ლუწი მთელი რიცხვებია (ყოველი $0 \leq i \leq n-1$-თვის).
- შადრევნების მდებარეობები განსხვავებულია.

ქვეამოცანები

1. (5 ქულა) $x[i] = 2$ (ყოველი $0 \leq i \leq n-1$-თვის)
2. (10 ქულა) $2 \leq x[i] \leq 4$ (ყოველი ($0 \leq i \leq n-1$)-თვის)
3. (15 ქულა) $2 \leq x[i] \leq 6$ (ყოველი ($0 \leq i \leq n-1$)-თვის)
4. (20 ქულა) არსებობს გზების დაგების მხოლო ერთი გზა, რომლის შემდეგაც შესაძლებელია ნებისმიერ ორ შადრევანს შორის გადაადგილება ამ გზების დახმარებით.
5. (20 ქულა) არცერთი ოთხი შადრევანი არ წარმოადგენს რომელიმე 2×2 კვადრატის წვეროებს.
6. (30 ქულა) დამატებითი შეზღუდვების გარეშე.

სანიმუშო გრადერი

სანიმუშო გრადერი კითხულობს მონაცემებს შემდეგი ფორმატით:

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

სანიმუშო გრადერს გამოაქვს მონაცემები შემდეგი ფორმატით:

- სტრიქონი 1: `construct_roads`-ის მიერ დაბრუნებული მნიშვნელობა

თუ `construct_roads`-მა დააბრუნა 1 და `build(u, v, a, b)` იქნა გამოძახებული, გრადერი დამატებით ბეჭდავს:

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