



სუპერხეების შეერთება (supertrees)

ბაღები ყურესთან არის დიდი ბუნებრივი პარკი სინგაპურში. პარკში დგას n ცალი კოშკი, რომლებიც ცნობილი არიან სუპერხეების სახელით. ეს კოშკები გადანომრილია 0-დან $(n - 1)$ -მდე. ჩვენ უნდა ავაგოთ სიმრავლე **ნული ან მეტი** ხიდისაგან. ყოველი ხიდი შეაერთებს განსხვავებული კოშკების წყვილს და შესაძლებელია მასზე ვიმოდრაოთ **ორივე** მიმართულებით. ორი სხვადასხვა ხიდი არ შეიძლება აერთებდეს კოშკების ერთსა და იმავე წყვილს.

გზა x კოშკიდან y კოშკამდე ეწოდება ერთი ან რამდენიმე კოშკიდან შედგენილ მიმდევრობას, სადაც:

- მიმდევრობის პირველი ელემენტი არის x ,
- მიმდევრობის ბოლო ელემენტი არის y ,
- ყველა ელემენტი ამ მიმდევრობაში არის **განსხვავებული**,
- თანმიმდევრულად განლაგებული ყოველი ორი ელემენტი (კოშკი) ამ მიმდევრობაში შეერთებულია ხიდით.

შევნიშნოთ, რომ განსაზღვრების თანახმად არსებობს ზუსტად ერთი გზა კოშკიდან თავის თავამდე, ხოლო განსხვავებული გზების რაოდენობა i კოშკიდან j კოშკამდე ტოლია განსხვავებული გზების რაოდენობისა j კოშკიდან i კოშკამდე.

წამყვან არქიტექტორს სურს ისე ააგოს ხიდები, რომ ყველა $0 \leq i, j \leq n - 1$ წყვილისათვის არსებობდეს ზუსტად $p[i][j]$ განსხვავებული გზა i კოშკიდან j კოშკამდე, სადაც $0 \leq p[i][j] \leq 3$.

ააგეთ ხიდების სიმრავლე, რომელიც აკმაყოფილებს არქიტექტორის მოთხოვნებს, ან დაადგინეთ, რომ ამის გაკეთება შეუძლებელია.

იმპლემენტაციის დეტალები

თქვენ უნდა მოახდინოთ შემდეგი პროცედურის იმპლემენტაცია:

```
int construct(int[][] p)
```

- p : $n \times n$ ზომის მასივი წარმოადგენს არქიტექტორის მოთხოვნებს.
- თუკი აგება შესაძლებელია, ამ პროცედურამ ზუსტად ერთხელ უნდა გამოიძახოს `build` (იხილე ქვემოთ) რომელიც გამოიტანს შეტყობინებას აგების შესახებ, რის შემდეგაც უნდა დააბრუნოს 1.
- წინააღმდეგ შემთხვევაში პროცედურამ უნდა დააბრუნოს 0, `build`-ის

გამოძახების გარეშე.

- ეს პროცედურა გამოიძახება ზუსტად ერთხელ.

პროცედურა `build` განისაზღვრება შემდეგნაირად:

```
void build(int[] [] b)
```

- b : $n \times n$ ზომის მასივი, სადაც $b[i][j] = 1$, თუ ხიდი აერთებს i და j კოშკებს, ან $b[i][j] = 0$ წინააღმდეგ შემთხვევაში.
- მიაქციეთ ყურადღება მასივი უნდა აკმაყოფილებდეს პირობას $b[i][j] = b[j][i]$ ყველა $0 \leq i, j \leq n - 1$ და $b[i][i] = 0$ ყველა $0 \leq i \leq n - 1$.

მაგალითები

მაგალითი 1

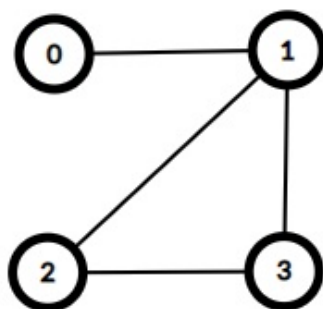
განვიხილოთ შემდეგი გამოძახება:

```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

ეს ნიშნავს, რომ უნდა არსებობდეს ზუსტად ერთი გზა 0 ნომრის მქონე კოშკიდან 1 ნომრის მქონე კოშკამდე. ყველა დანარჩენი (x, y) კოშკების წყვილისათვის, სადაც $0 \leq x < y \leq 3$, უნდა არსებობდეს ზუსტად ორი გზა x კოშკიდან y კოშკამდე. ამის მიღწევა შესაძლებელია 4 ხიდით, რომლებიც შეაერთებენ კოშკთა წყვილებს: $(0, 1)$, $(1, 2)$, $(1, 3)$ და $(2, 3)$.

ამ ამოხსნის შესატყობინებლად `construct` პროცედურამ უნდა მოახდინოს შემდეგი გამოძახება:

- `build([[0, 1, 0, 0], [1, 0, 1, 1], [0, 1, 0, 1], [0, 1, 1, 0]])`



ამის შემდეგ მან უნდა დააბრუნოს 1.

ამ შემთხვევაში არსებობს რამდენიმე კონსტრუქცია, რომელიც მოთხოვნებს აკმაყოფილებს. ყველა მათგანი ჩაითვლება სწორად.

მაგალითი 2

განვიხილოთ შემდეგი გამოცხება:

```
construct([[1, 0], [0, 1]])
```

ეს ნიშნავს, რომ არ არსებობს გზა ორ კოშკს შორის. ეს შესაძლებელია მხოლოდ ხიდების არარსებობის შემთხვევაში.

შესაბამისად, `construct` პროცედურამ უნდა მოახდინოს შემდეგი გამოცხება:

- `build([[0, 0], [0, 0]])`

ამის შემდეგ `construct` პროცედურამ უნდა დააბრუნოს 1-ს.

მაგალითი 3

განვიხილოთ შემდეგი გამოცხება:

```
construct([[1, 3], [3, 1]])
```

ეს ნიშნავს, რომ უნდა არსებობდეს ზუსტად 3 გზა 0 კოშკიდან 1 კოშკამდე. ამ მოთხოვნის დაკმაყოფილება შეუძლებელია, ამიტომ `construct` პროცედურამ უნდა დააბრუნოს 0 `build`-ის გამოცხების გარეშე.

შეზღუდვები

- $1 \leq n \leq 1000$
- $p[i][i] = 1$ ($0 \leq i \leq n - 1$)
- $p[i][j] = p[j][i]$ ($0 \leq i, j \leq n - 1$)
- $0 \leq p[i][j] \leq 3$ ($0 \leq i, j \leq n - 1$)

ქვეამოცანები

1. (11 ქულა) $p[i][j] = 1$ ($0 \leq i, j \leq n - 1$)
2. (10 ქულა) $p[i][j] = 0$ or 1 ($0 \leq i, j \leq n - 1$)
3. (19 ქულა) $p[i][j] = 0$ or 2 ($i \neq j, 0 \leq i, j \leq n - 1$)
4. (35 ქულა) $0 \leq p[i][j] \leq 2$ ($0 \leq i, j \leq n - 1$) და არსებობს ერთი მაინც კონსტრუქცია, რომელიც პირობებს აკმაყოფილებს.
5. (21 ქულა) $0 \leq p[i][j] \leq 2$ ($0 \leq i, j \leq n - 1$)
6. (4 ქულა) დამატებითი შეზღუდვების გარეშე.

სანიმუშო გრაფერი

სანიმუშო გრაფერს შეაქვს მონაცემები შემდეგი ფორმატით:

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

სანიმუშო გრაფერს გამოაქვს მონაცემები შემდეგი ფორმატით:

- სტრიქონი 1: `construct`-ის მიერ დაბრუნებული მნიშვნელობა.

თუ `construct` აბრუნებს 1-ს, სანიმუშო გრაფერი დამატებით გამოიტანს:

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