

Torņu savienošana (supertrees)

Līča dārzi ir liels Singapūras dabas parks, kurā atrodas n torņi, kas tiek saukti par superkokiem. Šie torņi ir sanumurēti ar skaitļiem no 0 līdz $n - 1$. Parka arhitekts vēlas uzbūvēt **nenegatīvu** skaitu tiltu. Katrs tilts savieno atšķirīgu torņu pāri un pa to var pārvietoties **abos** virzienos. Nekādi divi tilti nedrīkst savienot vienu un to pašu torņu pāri.

Ceļu no torņa x līdz tornim y veido tāda viena vai vairāku torņu virkne, kurā:

- virknes pirmais elements ir x ,
- virknes pēdējais elements ir y ,
- visi virknes elementi ir **atšķirīgi**, un
- katrus divus secīgus virknes elementus (torņus) savieno tilts.

Ievērojiet, ka pēc šīs definīcijas, no katra torņa ir tieši viens ceļš pašam uz sevi un atšķirīgo ceļu skaits no torņa i līdz tornim j sakrīt ar atšķirīgo ceļu skaitu no torņa j līdz tornim i .

Arhitekts vēlas uzbūvēt tiltus tā, lai visiem $0 \leq i, j \leq n - 1$ būtu tieši $p[i][j]$ atšķirīgi ceļi no torņa i līdz tornim j , kur $0 \leq p[i][j] \leq 3$.

Izveidojiet tādu tiltu kopu, kas atbilstu arhitekta izvirzītajām prasībām, vai arī nosakiet, ka tās izpildīt nav iespējams.

Implementēšanas detaļas

Jums ir nepieciešams implementēt šādu funkciju:

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

- p : $n \times n$ elementu masīvs, kurā aprakstītas arhitekta prasības.
- Ja tiltu kopu ar aprakstītajām īpašībām izveidot ir iespējams, šai funkcijai tieši vienreiz jāizsauc procedūra `build` (skat. tālāk), kurā jāapraksta izveidotā kopa, un pēc tam jāatgriež vērtība 1.
- Pretējā gadījumā funkcijai jāatgriež 0, neveicot procedūras `build` izsaukumus.
- Šī funkcija tiks izsaukta tieši vienreiz.

Procedūra `build` ir definēta šādi:

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

- b : $n \times n$ elementu masīvs, kur $b[i][j] = 1$, ja ir tilts, kas savieno torņus i un j , vai $b[i][j] = 0$

pretējā gadījumā.

- levērojiet, ka masīvam visiem $0 \leq i, j \leq n - 1$ ir jābūt spēkā īpašībai $b[i][j] = b[j][i]$ un visiem $0 \leq i \leq n - 1$ jābūt spēkā $b[i][i] = 0$.

Piemēri

1. piemērs

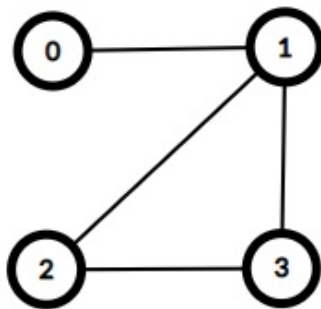
Aplūkosim šādu funkcijas izsaukumu:

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

Tas nozīmē, ka jābūt tieši vienam ceļam no torņa 0 līdz tornim 1. Visiem citiem torņu pāriem (x, y) , kur $0 \leq x < y \leq 3$, jābūt tieši diviem ceļiem no torņa x līdz tornim y . Šīs prasības apmierina četru tiltu kopa, kur tilti savieno torņu pārus $(0, 1)$, $(1, 2)$, $(1, 3)$ un $(2, 3)$.

Lai aprakstītu šo kopu, procedūra `construct` jāizsauc ar šādiem parametriem:

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



Pēc tam funkcijai jāatgriež vērtība 1.

Šajā gadījumā prasības apmierina vairākas tiltu kopas, un jebkura no tām tiks atzīta par pareizu.

2. piemērs

Aplūkosim šādu funkcijas izsaukumu:

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

Tas nozīmē, ka starp šiem diviem torņiem ceļam nav jābūt. To var panākt tikai neuzbūvējot nevienu tiltu.

Šajā gadījumā funkcijai `construct` jāizsauc procedūra `build` ar šādiem parametriem:

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

Pēc tam funkcijai `construct` jāatgriež vērtība 1.

3. piemērs

Aplūkosim šādu funkcijas izsaukumu:

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

Tas nozīmē, ka jābūt tieši 3 ceļiem no torņa 0 līdz tornim 1. Šīs prasības apmierināt nav iespējams, tāpēc funkcijai `construct` ir jāatgriež vērtība 0, pirms tam neveicot nevienu `build` izsaukumu.

Ierobežojumi

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

Apakšuzdevumi

1. (11 punkti) $p[i][j] = 1$ (visiem $0 \leq i, j \leq n - 1$)
2. (10 punkti) $p[i][j] = 0$ vai 1 (visiem $0 \leq i, j \leq n - 1$)
3. (19 punkti) $p[i][j] = 0$ vai 2 (visiem $i \neq j, 0 \leq i, j \leq n - 1$)
4. (35 punkti) $0 \leq p[i][j] \leq 2$ (visiem $0 \leq i, j \leq n - 1$) un ir vismaz viena torņu kopa, kas apmierina visas prasības.
5. (21 punkts) $0 \leq p[i][j] \leq 2$ (visiem $0 \leq i, j \leq n - 1$)
6. (4 punkti) Bez papildu ierobežojumiem.

Paraugvērtētājs

Paraugvērtētājs lasa ievaddatus šādā formātā:

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

Paraugvērtētājs izvada datus šādā formātā:

- 1. rinda: `construct` atgrieztā vērtība.

Ja `construct` atgrieztā vērtība ir 1, paraugvērtētājs papildus izdrukā:

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