

Supermedžių jungimas (supertrees)

"Gardens by the Bay" yra Singapūre esantis didelis gamtos parkas. Parke yra n bokštų, vadinamų supermedžiais. Šie bokštai sunumeruoti nuo 0 iki $n - 1$. Norime pastatyti **nulį arba daugiau** tiltų. Kiekvienas tiltas sujungia skirtingų bokštų porą ir gali būti naudojamas keliaujant **abiem** kryptimis. Jokie du tiltai neturėtų jungti tos pačios bokštų poros.

Kelias nuo x -ojo bokšto iki y -ojo bokšto yra vieno arba daugiau bokštų seka, kurioje:

- pirmasis sekos narys yra x ,
- paskutinis sekos narys yra y ,
- visi sekos nariai yra **skirtingi**, ir
- kiekvieni du iš eilės einantys sekos nariai (bokštai) yra sujungti tiltu.

Atkreipkite dėmesį, kad pagal apibrėžimą yra lygiai vienas kelias nuo kiekvieno bokšto iki jo paties, o skirtingų kelių skaičius nuo i -ojo bokšto iki j -ojo bokšto yra toks pats, kaip kelių skaičius nuo j -ojo bokšto iki i -ojo bokšto.

Vyriausiasis architektas norėtų tiltus statyti taip, kad visiems $0 \leq i, j \leq n - 1$ būtų lygiai $p[i][j]$ skirtingų kelių nuo i -ojo bokšto iki j -ojo bokšto, čia $0 \leq p[i][j] \leq 3$.

Sukonstruokite tiltų rinkinį, atitinkantį architekto reikalavimus, arba nustatykite, kad to padaryti neįmanoma.

Realizacija

Jums reikėtų parašyti šią funkciją:

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

- p : $n \times n$ dydžio dvimatis masyvas, nurodantis architekto reikalavimus.
- Jei konstrukcija yra įmanoma, ši procedūra turėtų lygiai vieną kartą iškviešti `build` (žiūrėti žemiau), kad perduotų konstrukciją, ir tada grąžinti 1.
- Kitu atveju funkcija turėtų grąžinti 0, neiškviesdama `build`.
- Ši funkcija iškviečiama lygiai vieną kartą.

Funkcija `build` apibrėžta taip:

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

- b : $n \times n$ dydžio dvimatis masyvas, kur $b[i][j] = 1$, jei yra tiltas, jungiantis i -ąjį ir j -ąjį bokštus, ir $b[i][j] = 0$ kitu atveju.
- Atkreipkite dėmesį, kad masyvas turi tenkinti $b[i][j] = b[j][i]$ visiems $0 \leq i, j \leq n - 1$ ir $b[i][i] = 0$ visiems $0 \leq i \leq n - 1$.

Pavyzdžiai

Pavyzdys 1

Panagrinėkime tokį funkcijos iškvietimą:

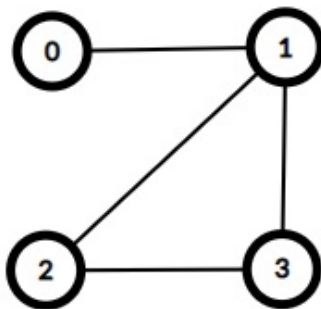
```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

Tai reiškia, kad turėtų būti lygiai vienas kelias nuo 0-inio bokšto iki 1-ojo bokšto. Visoms kitoms bokštų poroms (x, y) , kur $0 \leq x < y \leq 3$, turėtų būti lygiai du keliai nuo x -ojo bokšto iki y -ojo bokšto.

Tai galima pasiekti 4 tiltais, jungiančiais šias bokštų poras: $(0, 1)$, $(1, 2)$, $(1, 3)$ ir $(2, 3)$.

Kad perduotų sprendimą, funkcija `construct` turėtų atlikti šį iškvietimą:

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



Tada ji turėtų grąžinti 1.

Šiuo atveju yra daugiau nei viena konstrukcija, tenkinanti reikalavimus, ir visos tokios konstrukcijos laikomos teisingomis.

Pavyzdys 2

Panagrinėkime tokį funkcijos iškvietimą:

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

Tai reiškia, kad neturėtų būti galima keliauti tarp jokių dviejų bokštų. Tai galima pasiekti tik nestatant jokių tiltų.

Taigi funkcija `construct` turėtų atlikti šį iškvietaimą:

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

Tada funkcija `construct` turėtų grąžinti 1.

Pavyzdys 3

Panagrinėkime tokį funkcijos iškvietaimą:

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

Tai reiškia, kad turėtų būti lygiai 3 keliai nuo 0-inio bokšto iki 1-ojo bokšto. Šių reikalavimų įgyvendinti neįmanoma. Taigi funkcija `construct` turėtų grąžinti 0, neiškviesdama funkcijos `build`.

Ribojimai

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

Dalinės užduotys

1. (11 taškų) $p[i][j] = 1$ (visiems $0 \leq i, j \leq n - 1$)
2. (10 taškų) $p[i][j] = 0$ arba 1 (visiems $0 \leq i, j \leq n - 1$)
3. (19 taškų) $p[i][j] = 0$ arba 2 (visiems $i \neq j, 0 \leq i, j \leq n - 1$)
4. (35 taškai) $0 \leq p[i][j] \leq 2$ (visiems $0 \leq i, j \leq n - 1$) ir egzistuoja nors viena konstrukcija, tenkinanti reikalavimus.
5. (21 taškas) $0 \leq p[i][j] \leq 2$ (visiems $0 \leq i, j \leq n - 1$)
6. (4 taškai) Papildomų ribojimų nėra.

Pavyzdinė vertinimo programa

Pavyzdinė vertinimo programa perskaito įvestį šiuo formatu:

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

Pavyzdinės vertinimo programos išvestis pateikiama šiuo formatu:

- 1 – oji eilutė: funkcijos `construct` grąžinta vertė.

Jeigu funkcijos `construct` grąžinta vertė yra 1, pavyzdinė vertinimo programa papildomai išspausdina:

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