Divja Drevesa (supertrees)

Vrtovi ob Zalivu so velik naravni park v Singapurju. V parku je n stolpov, znanih kot super drevesa, za katera pravijo, da v živo izgledajo precej divje. Ti stolpi so označeni z indeksi 0 do $n-1$. Želimo konstruirati množico **nič ali več** mostov. Vsak most povezuje dva različna stolpa in ga lahko prehodimo v **obe** smeri. Istega para stolpov naj ne povezuje več mostov.

Pot od stolpa x do stolpa y je zaporedje enega ali več stolpov, tako da:

- prvi element zaporedja je x,
- zadnji element zaporedja je y,
- vsi elementi zaporedja se **razlikujejo**, in
- vsaka zaporedna elementa (stolpa) v zaporedju sta povezana z mostom.

Opomba: po definiciji obstaja natanko ena pot od stolpa do samega sebe in število različnih poti od stolpa i do stolpa j je enako številu različnih poti od stolpa j do stolpa i.

Vodilni arhitekt, zadolžen za projekt, želi, da bi bili mostovi zgrajeni tako, da za vsak $0 \leq i, j \leq n-1$, obstaja natanko $p[i][j]$ različnih poti od stolpa i do stolpa j, kjer velja $0 \leq p[i][j] \leq 3$.

Sestavi množico mostov, ki zadosti arhitektovim zahtevam, ali pa ugotovi, da je to nemogoče.

Podrobnosti impementacije

Implementiraj naslednjo funkcijo:

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

- p: $n \times n$ polje, ki predstavlja arhitektove zahteve.
- Če je takšno množico moč sestaviti, naj funkcija pokliče proceduro `build` (glej spodaj) natanko enkrat, za tem pa naj vrne 1.
- Sicer funkcija vrne 0, brez da kliče `build`.
- Ta funkcija je klicana natanko enkrat.

Procedura `build` je definirana kot:

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

- b: $n \times n$ polje, kjer je $b[i][j] = 1$, če obstaja most, ki povezuje stolp i in stolp j; sicer je

$b[i][j] = 0$.
- Opomba: Polje mora zadostiti $b[i][j] = b[j][i]$ za vsak $0 \leq i, j \leq n-1$ in $b[i][i] = 0$, za vsak $0 \leq i \leq n-1$.

Primeri

1. primer

Obravavamo naslednji klic:

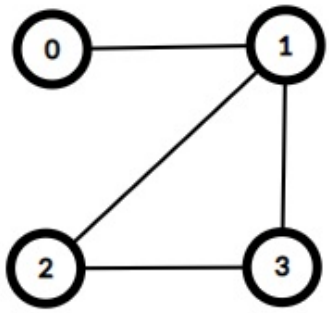
```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

To pomeni, da želimo natanko eno pot med stolpoma 0 in 1. Za vse ostale pare stolpov (x, y), kjer je $0 \leq x < y \leq 3$, naj bi obstajali natanko dve poti od stolpa x do stolpa y.

To lahko dosežemo s 4 mostovi, ki povezujejo pare stolpov $(0, 1)$, $(1, 2)$, $(1, 3)$ in $(2, 3)$.

Da sporočimo to rešitev, funkcija `construct` izvede naslednji klic:

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



Za tem funkcija vrne 1.

V tem primeru je več možnih rešitev, ki ustrezajo zahtevam, in vse izmed njih obravnavamo kot pravilne.

2. primer

Obravavamo naslednji klic:

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

To pomeni, da med dvema stolpoma ne sme biti poti. Temu lahko zadovoljimo le tako, da ne obstaja noben most.

Zatorej, funkcija `construct` izvede naslednji klic:

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

Za tem funkcija vrne 1.

3. primer

Obravavamo naslednji klic:

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

To pomeni, da pričakujemo natanko 3 poti od stolpa 0 do stolpa 1. Tem zahtevam ni moč zadostiti. V tem primeru funkcija `construct` vrne 0, brez da kliče proceduro `build`.

Omejitve

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

Podnaloge

1. (11 točk) $p[i][j] = 1$ (za vsak $0 \leq i, j \leq n-1$)
2. (10 točk) $p[i][j] = 0$ or 1 (za vsak $0 \leq i, j \leq n-1$)
3. (19 točk) $p[i][j] = 0$ or 2 (za vsak $i \neq j$, $0 \leq i, j \leq n-1$)
4. (35 točk) $0 \leq p[i][j] \leq 2$ (za vsak $0 \leq i, j \leq n-1$) in obstaja vsaj ena veljavna rešitev.
5. (21 točk) $0 \leq p[i][j] \leq 2$ (za vsak $0 \leq i, j \leq n-1$)
6. (4 točk) Ni dodatnih omejitev.

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod v naslednjem formatu:

- vrstica 1: n
- vrstice $2+i$ ($0 \leq i \leq n-1$): $p[i][0]$ $p[i][1]$ $\ldots$ $p[i][n-1]$

Izhod vzorčnega ocenjevalnika je v naslednjem formatu:

- vrstica 1: vrnjena vrednost klica `construct`.

Če klic `construct` vrne 1, vzorčni ocenjevalnik dodatno izpiše:

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