

## Spojování věží (supertrees)

Zahrady u zálivu (Gardens by the Bay) jsou rozsáhlým přírodním parkem v Singapuru. V parku je postaveno  $n$  věží, které místní obyvatelé nazývají "supertrees". Tyto věže jsou očíslovány od 0 do  $n - 1$ . Chtěli bychom mezi věžemi postavit několik (**nula nebo více**) mostů. Každý most spojuje dvojici různých věží a můžeme po něm chodit **oběma** směry. Žádné dva mosty nespojují stejnou dvojici věží.

Cestou od věže  $x$  k věži  $y$  nazveme takovou posloupnost jedné nebo více věží, která splňuje tyto podmínky:

- prvním prvkem této posloupnosti je  $x$ ,
- posledním prvkem této posloupnosti je  $y$ ,
- všechny prvky této posloupnosti jsou navzájem **různé**, a
- každé dva po sobě jdoucí prvky (věže) této posloupnosti jsou spojeny mostem.

Poznamenejme, že podle uvedené definice existuje právě jedna cesta od věže  $i$  k věži  $i$  (tzn. se stejným začátkem a koncem) a počet různých cest od věže  $i$  k věži  $j$  je stejný, jako počet různých cest od věže  $j$  k věži  $i$ .

Vedoucí architekt odpovědný za návrh požaduje postavit mosty tak, aby pro všechna  $0 \leq i, j \leq n - 1$  existovalo přesně  $p[i][j]$  různých cest od věže  $i$  k věži  $j$ , kde  $0 \leq p[i][j] \leq 3$ .

Sestrojte mosty tak, abyste vyhověli všem požadavkům architekta, nebo oznamte, že to není možné.

## Implementační detaily

Implementujte následující funkce:

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

- $p$ : pole velikosti  $n \times n$  obsahující požadavky architekta.
- Je-li možné postavit mosty podle zadaných požadavků architekta, funkce provede právě jedno volání `build` (viz níže), kterým rozmístění mostů popíše; následně vrátí 1.
- V opačném případě funkce vrátí 0, aniž by zavolala `build`.
- Tato funkce bude zavolána právě jednou.

Funkce `build` je definována takto:

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

- $b$ : pole velikosti  $n \times n$  s hodnotami  $b[i][j] = 1$ , jestliže má být postaven most spojující věž  $i$  a věž  $j$ , nebo  $b[i][j] = 0$  v opačném případě.
- Poznamenejme, že hodnoty pole  $b$  musí splňovat podmínky  $b[i][j] = b[j][i]$  pro všechna  $0 \leq i, j \leq n - 1$  a  $b[i][i] = 0$  pro všechna  $0 \leq i \leq n - 1$ .

## Příklady

### Příklad 1

Uvažujme následující volání:

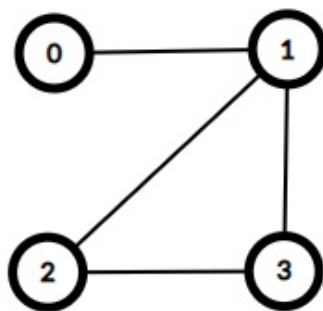
```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

Znamená to, že musí existovat právě jedna cesta mezi věží 0 a věží 1. Pro všechny ostatní dvojice věží  $(x, y)$  takové, že  $0 \leq x < y \leq 3$ , musí existovat právě dvě různé cesty od věže  $x$  k věži  $y$ .

Tyto požadavky lze splnit pomocí 4 mostů, které budou spojovat dvojice věží  $(0, 1)$ ,  $(1, 2)$ ,  $(1, 3)$  a  $(2, 3)$ .

Funkce `construct` oznámí tento výsledek následujícím voláním:

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



Potom vrátí hodnotu 1.

V tomto případě má úloha více možných řešení, která odpovídají zadaným požadavkům. Kterékoli z nich bude vyhodnoceno jako správné.

### Příklad 2

Uvažujme následující volání:

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

Znamená to, že mezi oběma věžemi nesmí existovat žádná cesta. To lze zajistit tím, že nepostavíme žádný most.

Proto funkce `construct` provede volání:

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

Následně funkce `construct` vrátí 1.

### Příklad 3

Uvažujme následující volání:

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

Znamená to, že mají existovat 3 různé cesty mezi věžemi 0 a 1. Takový požadavek ovšem nelze splnit. Proto funkce `construct` vrátí hodnotu 0, aniž by předtím zavolala `build`.

## Omezení

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

## Podúlohy

1. (11 bodů)  $p[i][j] = 1$  (pro všechna  $0 \leq i, j \leq n - 1$ )
2. (10 bodů)  $p[i][j] = 0$  nebo 1 (pro všechna  $0 \leq i, j \leq n - 1$ )
3. (19 bodů)  $p[i][j] = 0$  nebo 2 (pro všechna  $i \neq j, 0 \leq i, j \leq n - 1$ )
4. (35 bodů)  $0 \leq p[i][j] \leq 2$  (pro všechna  $0 \leq i, j \leq n - 1$ ) a vždy existuje alespoň jedno vyhovující řešení
5. (21 bodů)  $0 \leq p[i][j] \leq 2$  (pro všechna  $0 \leq i, j \leq n - 1$ )
6. (4 body) Žádná další omezení.

## Ukázkový vyhodnocovač

Ukázkový vyhodnocovač čte vstup v následujícím tvaru:

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

Výstup ukázkového vyhodnocovače má tento formát:

- řádek 1: návratová hodnota funkce `construct`.

Je-li návratová hodnota funkce `construct` rovna 1, ukázkový vyhodnocovač dále vypíše:

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