

## Поврзување на супер дрва (supertrees)

„Градините покрај заливот“ е голем природен парк во Сингапур. Во паркот има  $n$  кули, познати како супер дрва. Овие кули се означени од 0 до  $n - 1$ . Ние би сакале да конструираме множество од **нула или повеќе** мостови. Секој мост поврзува пар од различни кули и може по него да се поминува во **било која** насока. Два моста не смеат да поврзуваат ист пар од кули.

Патека од кулата  $x$  до кулата  $y$  е секвенца од една или повеќе кули така што:

- првиот елемент од секвенцата е  $x$ ,
- последниот елемент од секвенцата е  $y$ ,
- сите елементи од секвенцата се **различни**, и
- секои два последователни елементи (кули) во секвенцата се поврзани со мост.

Забележете дека по дефиниција постои точно една патека од кула до самата себе и дека бројот на различни патеки од кулата  $i$  до кулата  $j$  е ист како и бројот на различни патеки од кулата  $j$  до кулата  $i$ .

Водечкиот архитект одговорен за дизајнот посакува мостовите да бидат изградени така што за сите  $0 \leq i, j \leq n - 1$  има точно  $p[i][j]$  различни патеки од кулата  $i$  до кулата  $j$ , каде  $0 \leq p[i][j] \leq 3$ .

Конструирајте множество од мостови кои ги задоволуваат барањата на архитектот, или пак утврдете дека тоа е невозможно.

## Имплементациски детали

Потребно е да ја имплементирате следната процедура:

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

- $p$ :  $n \times n$  низа која ги претставува барањата на архитектот.
- Ако е можна конструкција, оваа процедура треба да направи точно еден повик до `build` (видете подолу) за да ја пријави конструкцијата, по што треба да врати 1.
- Во спротивно, процедурата треба да врати 0 без да прави било какви повици до `build`.
- Оваа процедура се повикува точно еднаш.

Процедурата `build` е дефинирана на следниот начин:

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

- $b$ :  $n \times n$  низа, со  $b[i][j] = 1$  ако има мост што ги поврзува кулата  $i$  и кулата  $j$ , или  $b[i][j] = 0$  инаку.
- Забележете дека низата мора да го задоволи следното:  $b[i][j] = b[j][i]$  за сите  $0 \leq i, j \leq n - 1$  и  $b[i][i] = 0$  за сите  $0 \leq i \leq n - 1$ .

## Примери

### Пример 1

Земете го за пример следниот повик:

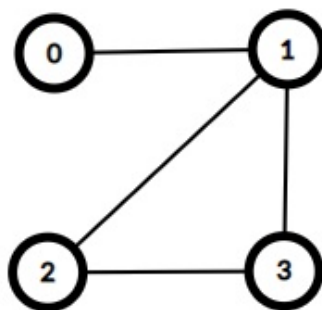
```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

Ова значи дека треба да има точно една патека од кулата 0 до кулата 1. За сите други парови од кули  $(x, y)$ , така што  $0 \leq x < y \leq 3$ , треба да има точно две патеки од кулата  $x$  до кулата  $y$ .

Ова може да се постигне со 4 моста, кои ги поврзуваат следните парови од кули:  $(0, 1)$ ,  $(1, 2)$ ,  $(1, 3)$  и  $(2, 3)$ .

За пријавување на ова решение, `construct` процедурата треба да го направи следниот повик:

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



Потоа треба да врати 1.

Во овој случај, постојат повеќе конструкции што одговараат на барањата, од кои сите ќе се сметаат за точни.

### Пример 2

Земете го за пример следниот повик:

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

Ова значи дека не постои начин да се патува помеѓу двете кули. Ова може да биде задоволено само доколу не постојат мостови.

Според тоа, `construct` процедурата треба да го направи следниот повик:

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

После овој повик, `construct` процедурата треба да врати 1.

### Пример 3

Земете го за пример следниот повик:

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

Ова значи дека треба да има точно 3 патеки од кулата 0 до кулата 1. Ова множество од барања не може да биде задоволено. Како такво, `construct` процедурата треба да врати 0 без да прави повици до `build`.

## Ограничувања

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

## Подзадачи

1. (11 поени)  $p[i][j] = 1$  (за сите  $0 \leq i, j \leq n - 1$ )
2. (10 поени)  $p[i][j] = 0$  или 1 (за сите  $0 \leq i, j \leq n - 1$ )
3. (19 поени)  $p[i][j] = 0$  или 2 (за сите  $i \neq j, 0 \leq i, j \leq n - 1$ )
4. (35 поени)  $0 \leq p[i][j] \leq 2$  (за сите  $0 \leq i, j \leq n - 1$ ) и постои најмалку една конструкција која ги задоволува барањата.
5. (21 поени)  $0 \leq p[i][j] \leq 2$  (за сите  $0 \leq i, j \leq n - 1$ )
6. (4 поени) Нема дополнителни ограничувања.

## Пример оценувач

Пример оценувачот го чита влезот во следниот формат:

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

Излезот од пример оценувачот е во следниот формат:

- линија 1: вредноста која се враќа од `construct`.

Ако вредноста која ја враќа `construct` е 1, пример оценувачот дополнително печати:

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