Свързване на супердървета (supertrees)

Градините на Бей е голям природен парк в Сингапур. В парка има n кули, познати като супердървета. Тези кули са номерирани от 0 до $n-1$. Бихме искали да построим множество от **нула или повече** моста. Всеки мост свързва две различни кули и може да се обхожда във **всяка** посока. Не трябва да има два моста, свързващи едни и същи кули.

Път от кула x до кула y е редица от една или повече кули, такива че:

- първият елемент на редицата е x,
- последният елемент на редицата е y,
- всички елементи на редицата са **различни**, и
- всеки два съседни елемента (кули) в редицата са свързани с мост.

Забележете, че по дефиниция има точно един път от една кула до себе си и броя различни пътища от кула i до кула j е равен на броя различни пътища от кула j до кула i.

Главният архитект, отговорен за проекта, иска мостовете да бъдат построени така, че за всяко $0 \leq i, j \leq n-1$ да има точно $p[i][j]$ различни пътя от кула i до кула j, където $0 \leq p[i][j] \leq 3$.

Намерете множество от мостове, които изпълняват изискванията на архитекта или определете, че това е невъзможно.

Имплементация

Трябва да напишете следната функция:

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

- p: масив $n \times n$ представляващ изискванията на архитекта.
- Ако конструкцията е възможна, тази функция трябва да извика точно един път `build` (виж по-долу) да съобщи решението, след което трябва да върне 1.
- В противен случай, функцията трябва да върне 0 без да прави никакви извиквания към `build`.
- Тази функция ще бъде извикана точно веднъж.

Функцията `build` е дефинирана както следва:

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

- b: масив $n \times n$, където $b[i][j] = 1$ ако има мост свързващ кулите i и j, или $b[i][j] = 0$ в противен случай.
- Забележете, че трябва $b[i][j] = b[j][i]$ за всяко $0 \leq i, j \leq n - 1$ и $b[i][i] = 0$ за всяко $0 \leq i \leq n - 1$.

Примери

Пример 1

Разглеждаме следното извикване:

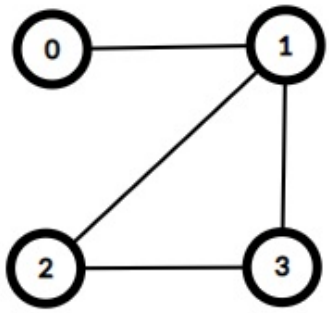
```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

Това значи, че трябва да има точно един път от кула 0 до кула 1. За всяка друга двойка кули (x, y), такава че $0 \leq x < y \leq 3$, трябва да има точно два пътя от кула x до кула y.

Това може да се постигне с 4 моста, свързвайки двойките кули $(0, 1)$, $(1, 2)$, $(1, 3)$ и $(2, 3)$.

Да съобщи за това решение, функцията `construct` трябва да направи следното извикване:

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



След което трябва да върне 1.

В този случай има няколко множества мостове, които изпълняват изискванията, и всяко от тях ще бъде прието за вярно.

Пример 2

Разглеждаме следното извикване:

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

Това означава, че не трябва да има път между двете кули. Това може да се постигне само ако няма мостове.

Следователно, функцията `construct` трябва да направи следното извикване:

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

След което, функцията `construct` трябва да върне 1.

Пример 3

Разглеждаме следното извикване:

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

Това означава, че трябва да има точно 3 пътя от кула 0 до кула 1. Тези изисквания не могат да бъдат изпълнени. За това, функцията `construct` трябва да върне 0 без да прави извиквания на `build`.

Ограничения

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

Подзадачи

1. (11 точки) $p[i][j] = 1$ (за всяко $0 \leq i, j \leq n-1$)
2. (10 точки) $p[i][j] = 0$ или 1 (за всяко $0 \leq i, j \leq n-1$)
3. (19 точки) $p[i][j] = 0$ или 2 (за всяко $i \neq j$, $0 \leq i, j \leq n-1$)
4. (35 точки) $0 \leq p[i][j] \leq 2$ (за всяко $0 \leq i, j \leq n-1$) и има поне едно множество от мостове отговарящо на изискванията.
5. (21 точки) $0 \leq p[i][j] \leq 2$ (за всяко $0 \leq i, j \leq n-1$)
6. (4 точки) Няма допълнителни ограничения.

Примерен грейдър

Примерният грейдър чете входа в следния формат:

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

Примерният грейдър отпечатва изхода в следния формат:

- ред 1: върнатата стойност от `construct`.

Ако върнатата стойност от `construct` е 1, примерният грейдър допълнително отпечатва:

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