

Conectando Supertrees (supertrees)

Jardins da Baía é um grande parque natural em Singapura. No parque existem n torres, conhecidas como supertrees. Essas torres são rotuladas de 0 a $n - 1$. Nós gostaríamos de construir um conjunto de **zero ou mais** pontes. Cada ponte conecta um par de torres distintas e pode ser percorrida nas **duas** direções. Nenhum par de pontes deve conectar o mesmo par de torres.

Um caminho da torre x para a torre y é uma sequência de uma ou mais torres tais que:

- o primeiro elemento da sequência é x ,
- o último elemento da sequência é y ,
- todos os elementos da sequência são **distintos**, e
- cada dois elementos consecutivos (torres) na sequência são conectados por uma ponte.

Note que por definição há exatamente um caminho de uma torre para ela mesma e o número de caminhos diferentes da torre i para a torre j é o mesmo número de caminhos diferentes da torre j para a torre i .

O arquiteto líder responsável pelo projeto deseja que as pontes sejam construídas de tal forma que para todo $0 \leq i, j \leq n - 1$ exista exatamente $p[i][j]$ caminhos diferentes da torre i para a torre j , onde $0 \leq p[i][j] \leq 3$.

Construa um conjunto de pontes que satisfaça os requisitos do arquiteto, ou determine que isso é impossível.

Detalhes de implementação

Você deve implementar o seguinte procedimento:

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

- p : um array $n \times n$ representando os requisitos do arquiteto.
- Se uma construção for possível, esse procedimento deve realizar exatamente uma chamada a `build` (veja abaixo) para reportar a construção, em seguida deve retornar 1.
- Senão, o procedimento deve retornar 0 sem fazer nenhuma chamada a `build`.
- Esse procedimento é chamado exatamente uma vez.

O procedimento `build` é definido a seguir:

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

- b : um array $n \times n$, com $b[i][j] = 1$ se existe uma ponte conectando a torre i e a torre j , ou $b[i][j] = 0$ caso contrário.
- Note que o array deve satisfazer $b[i][j] = b[j][i]$ para todo $0 \leq i, j \leq n - 1$ e $b[i][i] = 0$ para todo $0 \leq i \leq n - 1$.

Exemplos

Exemplo 1

Considere a seguinte chamada:

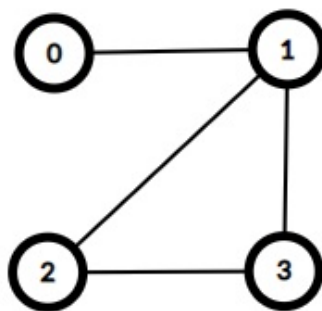
```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

Isso significa que deve existir exatamente um caminho da torre 0 para a torre 1. Para todos os outros pares de torres (x, y) , tais que $0 \leq x < y \leq 3$, devem existir exatamente dois caminhos da torre x para a torre y .

Isso pode ser realizado com 4 pontes, conectando os pares de torres $(0, 1)$, $(1, 2)$, $(1, 3)$ e $(2, 3)$.

Para reportar esta solução, o procedimento `construct` deve realizar a seguinte chamada:

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



E depois deve retornar 1.

Neste caso, existem múltiplas construções que satisfazem as restrições, todas elas seriam consideradas corretas.

Exemplo 2

Considere a seguinte chamada:

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

Isso significa que não deve existir forma de viajar entre as duas torres. Isso só pode ser satisfeito não havendo nenhuma ponte.

Então, o procedimento `construct` deve realizar a seguinte chamada:

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

Depois disso, o procedimento `construct` deve retornar 1.

Exemplo 3

Considere a seguinte chamada:

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

Isso significa que devem existir exatamente 3 caminhos da torre 0 para a torre 1. Esse conjunto de restrições não pode ser satisfeito. Assim sendo, o procedimento `construct` deve retornar 0 sem realizar nenhuma chamada a `build`.

Restrições

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

Subtarefas

1. (11 pontos) $p[i][j] = 1$ (para todo $0 \leq i, j \leq n - 1$)
2. (10 pontos) $p[i][j] = 0$ ou 1 (para todo $0 \leq i, j \leq n - 1$)
3. (19 pontos) $p[i][j] = 0$ ou 2 (para todo $i \neq j, 0 \leq i, j \leq n - 1$)
4. (35 pontos) $0 \leq p[i][j] \leq 2$ (para todo $0 \leq i, j \leq n - 1$) e existe pelo menos uma construção satisfazendo as restrições.
5. (21 pontos) $0 \leq p[i][j] \leq 2$ (para todo $0 \leq i, j \leq n - 1$)
6. (4 pontos) Nenhuma restrição adicional.

Corretor exemplo

O corretor exemplo lê a entrada no seguinte formato:

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

A saída do corretor exemplo está no seguinte formato:

- linha 1: o valor de retorno de `construct`.

Se o valor de retorno de `construct` é 1, o corretor exemplo adicionalmente imprime:

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