

Menghubungkan Supertrees (supertrees)

Gardens by the Bay adalah sebuah taman alam yang luas di Singapura. Di taman tersebut terdapat n buah menara, yang lebih dikenal dengan nama *supertrees*. Menara ini dilabeli dari 0 sampai $n - 1$. Kita ingin membangun sebuah set yang terdiri dari **nol atau lebih** jembatan. Tiap jembatan menghubungkan sebuah pasang menara berbeda dan dapat dilewati dari **kedua arah**. Tidak ada dua jembatan yang menghubungkan pasangan menara yang sama.

Sebuah jalur dari menara x ke menara y adalah sebuah barisan dari satu atau lebih menara sedemikian sehingga:

- elemen pertama dari barisan tersebut adalah x ,
- elemen terakhir dari barisan tersebut adalah y ,
- semua elemen pada barisan tersebut **berbeda-beda**, dan
- tiap dua elemen berurutan (menara) di barisan tersebut terhubung oleh sebuah jembatan.

Perhatikan bahwa berdasarkan definisi, terdapat tepat satu jalur dari tiap menara ke dirinya sendiri dan banyaknya jalur dari menara i ke menara j sama dengan banyaknya jalur dari menara j ke menara i .

Pemimpin arsitektur yang bertanggung jawab atas desain berharap agar jembatan dibangun sehingga untuk setiap $0 \leq i, j \leq n - 1$ terdapat tepat $p[i][j]$ jalur berbeda dari menara i ke menara j , dimana $0 \leq p[i][j] \leq 3$.

Buatlah sebuah set jembatan yang memenuhi permintaan arsitek, atau tentukan apabila hal itu tidak mungkin.

Detail Implementasi

Anda harus mengimplementasikan fungsi berikut:

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

- p : Sebuah array berukuran $n \times n$ menyatakan permintaan arsitek.
- Apabila terdapat konstruksi yang mungkin, fungsi ini harus melakukan tepat satu pemanggilan ke `build` (lihat dibawah) untuk melaporkan konstruksi itu, kemudian fungsi ini harus mengembalikan 1.
- Selain itu, fungsi ini harus mengembalikan 0 tanpa melakukan pemanggilan apapun ke `build`.
- fungsi ini dipanggil tepat sekali.

fungsi `build` didefinisikan sebagai berikut:

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

- b : Sebuah array berukuran $n \times n$, dengan $b[i][j] = 1$ apabila terdapat sebuah jembatan yang menghubungkan menara i dan menara j , atau $b[i][j] = 0$ bila tidak.
- Perhatikan bahwa array itu harus memenuhi $b[i][j] = b[j][i]$ untuk setiap $0 \leq i, j \leq n - 1$ dan $b[i][i] = 0$ untuk setiap $0 \leq i \leq n - 1$.
- Note that the array must satisfy $b[i][j] = b[j][i]$ for all $0 \leq i, j \leq n - 1$ and $b[i][i] = 0$ for all $0 \leq i \leq n - 1$.

Contoh

Contoh 1

Perhatikan pemanggilan berikut:

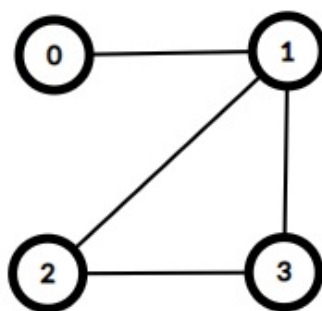
```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

Hal ini berarti terdapat tepat satu jalur dari menara 0 ke menara 1. Untuk setiap pasangan menara yang lain (x, y) , dengan $0 \leq x < y \leq 3$, terdapat tepat dua jalur dari menara x ke menara y .

Hal ini dapat dicapai dengan 4 jembatan, yang menghubungkan pasangan menara $(0, 1)$, $(1, 2)$, $(1, 3)$ dan $(2, 3)$.

Untuk melaporkan solusi ini, fungsi `construct` harus melakukan pemanggilan berikut:

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



fungsi kemudian harus mengembalikan 1.

Di kasus ini, terdapat beberapa konstruksi yang memenuhi permintaan, semua konstruksi tersebut dianggap benar.

Contoh 2

Perhatikan pemanggilan berikut:

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

Hal ini berarti tidak ada cara untuk berpindah antara kedua menara. Hal ini hanya dapat dipenuhi dengan tidak memiliki jembatan sama sekali.

Karena itu, fungsi `construct` harus melakukan pemanggilan berikut:

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

Kemudian, fungsi `construct` harus mengembalikan 1.

Contoh 3

Perhatikan pemanggilan berikut:

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

Hal ini berarti terdapat tepat 3 jalur dari menara 0 ke menara 1. Persyaratan ini tidak dapat dipenuhi. Oleh karena itu, fungsi `construct` harus mengembalikan 0 tanpa melakukan pemanggilan ke `build`.

Batasan

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

Subsoal

1. (11 poin) $p[i][j] = 1$ (untuk setiap $0 \leq i, j \leq n - 1$)
2. (10 poin) $p[i][j] = 0$ or 1 (untuk setiap $0 \leq i, j \leq n - 1$)
3. (19 poin) $p[i][j] = 0$ or 2 (untuk setiap $i \neq j, 0 \leq i, j \leq n - 1$)
4. (35 poin) $0 \leq p[i][j] \leq 2$ (untuk setiap $0 \leq i, j \leq n - 1$) dan terdapat paling tidak satu konstruksi yang memenuhi permintaan.
5. (21 poin) $0 \leq p[i][j] \leq 2$ (untuk setiap $0 \leq i, j \leq n - 1$)
6. (4 poin) Tidak ada batasan tambahan.

Contoh *grader*

Contoh *grader* akan membaca masukan dengan format berikut:

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

Contoh *grader* akan mencetak jawaban Anda dengan format berikut:

- baris 1: nilai yang dikembalikan `construct`.

Apabila nilai yang dikembalikan `construct` adalah 1, contoh *grader* akan mencetak tambahan keluaran:

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