Süperağaçları Bağlama (supertrees)

Gardens by the Bay Singapur'da büyük bir doğa parkıdır. Parkta süperağaçlar olarak bilinen n tane kule vardır. Bu kuleler 0'dan $n-1$'e kadar numaralandırılmıştır. Biz **sıfır ya da daha fazla** köprü kurmak istiyoruz. Her köprü birbirinden farklı iki kuleyi birbirine bağlamakta olup **her iki yönde de** kullanılabilir. İki kuleyi birbirine bağlayan birden fazla köprü olamaz.

x kulesinden y kulesine bir yol aşağıdaki özellikleri sağlayan bir kule dizisi olarak tanımlanmıştır:

- dizinin ilk elemanı x'dir,
- dizinin son elemanı y'dir,
- dizinin bütün elemanları birbirinden **farklıdır**, ve
- dizinin her ardışık iki elemanı (kule) birbirlerine bir köprüyle bağlıdır.

Dikkat edilirse tanım gereği her kuleden kendisine sadece ve sadece bir yol vardır. Her i ve j için, i kulesinden j kulesine farklı yol sayısı j kulesinden i kulesine farklı yol sayısına eşittir.

Köprüleri tasarlamaktan sorumlu baş mimar $0 \leq i, j \leq n-1$ olmak üzere her i ve j sayısı için i kulesinden j kulesine tam olarak $p[i][j]$ farklı yol olmasını istemektedir. Her i ve j için $0 \leq p[i][j] \leq 3$'tür.

Sizden ya mimarın isteklerine uygun olarak köprüler inşa etmeniz ya da bunun mümkün olmadığını belirlemeniz istenmektedir.

Implementasyon detayları

Aşağıdaki fonksiyonu implement etmelisiniz:

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

- p: Mimarın isteklerini gösteren $n \times n$'lik bir dizi.
- Köprüleri mimarın isteklerine uygun şekilde inşa etmek mümkünse, bu fonksiyon köprülerin inşa edileceği yerleri bildirmek için aşağıda tanımlanan `build` fonksiyonunu tam olarak bir kere çağıracaktır. Sonrasında da 1 döndürecektir.
- Aksi taktirde, `build` fonksiyonu çağırılmayacak, sadece 0 döndürülecektir.
- `construct` fonksiyonu sadece bir kere çağırılacaktır.

`build` fonksiyonu aşağıdaki gibi tanımlanmıştır:

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

- b: $n \times n$'lik bir dizidir. i kulesini j kulesine bağlayan bir köprü varsa $b[i][j] = 1$, aksi taktirde $b[i][j] = 0$'dır.
- Dikkat edilirse $0 \leq i, j \leq n - 1$ olmak üzere her i ve j için $b[i][j] = b[j][i]$ eşitliği sağlanmalıdır. Benzer şekilde $0 \leq i \leq n - 1$ olmak üzere her i için $b[i][i] = 0$ olmalıdır.

Örnekler

Örnek 1

Aşağıdaki çağrıya bakınız:

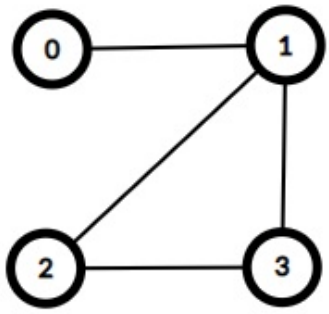
```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

Bu örnekte 0 kulesinden 1 kulesine sadece bir tane yol olmalıdır. $0 \leq x < y \leq 3$ olmak üzere her (x, y) ikilisi için x kulesinden y kulesine tam olarak iki yol olmalıdır.

Bu şartlar $(0, 1)$, $(1, 2)$, $(1, 3)$ ve $(2, 3)$ kulelerini bağlayan 4 köprü ile gerçekleştirilebilir.

Çözümü bildirmek için `construct` fonksiyonu aşağıdaki çağrıyı yapıp, sonrasında ise 1 döndürmelidir.

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



Bu örnekte şartları sağlayan birden çok çözüm vardır ve hepsi de doğru kabul edilecektir.

Örnek 2

Aşağıdaki çağrıya bakınız:

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

Bu örnekte iki kule arasında hiçbir yol olmamalıdır. Bu şart sadece hiç köprü inşa etmeyerek sağlanabilir.

Bu yüzden, `construct` fonksiyonu aşağıdaki çağrıyı yapıp, sonrasında ise 1 döndürmelidir.

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

Örnek 3

Aşağıdaki çağrıya bakınız:

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

Bu örnekte kule 0 ve kule 1 arasında tam olarak 3 yol olmalıdır. Bu şartı sağlamak mümkün değildir. Bu yüzden `construct` fonksiyonu `build` fonksiyonunu çağırmamalı ve 0 döndürmelidir.

Kısıtlar

- $1 \leq n \leq 1000$
- $p[i][i] = 1$ ($0 \leq i \leq n-1$ olmak üzere her i için)
- $p[i][j] = p[j][i]$ ($0 \leq i, j \leq n-1$ olmak üzere her i ve j için)
- $0 \leq p[i][j] \leq 3$ ($0 \leq i, j \leq n-1$ olmak üzere her i ve j için)

Altgörevler

1. (11 puan) $p[i][j] = 1$ ($0 \leq i, j \leq n-1$ olmak üzere her i ve j için)
2. (10 puan) $p[i][j] = 0$ or 1 ($0 \leq i, j \leq n-1$ olmak üzere her i ve j için)
3. (19 puan) $p[i][j] = 0$ or 2 ($i \neq j$ ve $0 \leq i, j \leq n-1$ olmak üzere her i ve j için)
4. (35 puan) $0 \leq p[i][j] \leq 2$ ($0 \leq i, j \leq n-1$ olmak üzere her i ve j için) ve şartları sağlayan en az bir çözüm vardır.
5. (21 puan) $0 \leq p[i][j] \leq 2$ ($0 \leq i, j \leq n-1$ olmak üzere her i ve j için)
6. (4 puan) Başka kısıt yoktur.

Örnek Grader

Örnek grader girdiyi aşağıdaki formatta okumaktadır:

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

Örnek grader çıktısı aşağıdaki formattadır:

- satır 1: `construct` fonksiyonunun döndürdüğü değer.

Eğer `construct` fonksiyonu 1 döndürüyorsa, örnek grader ek olarak aşağıdakileri de yazmaktadır:

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