

Toy Design

Problem Adı	ToyDesign
Girdi Dosyası	Etkileşimli (İnteraktif) Görev
Çıktı Dosyası	Etkileşimli (İnteraktif) Görev
Zaman limiti	1 saniye
Bellek limiti	256 megabytes

Oyuncak tasarlayan bir şirkette çalışıyorsunuz. Oluşturulmakta olan yeni bir oyuncak şu şekilde çalışmaktadır: Bir kutudan dışarı çıkan, 1 ile n arasında numaralandırılmış n adet iğne vardır. Bazı iğne çiftleri kutunun içindeki tellerle birbirine bağlıdır. (Başka bir deyişle, iğneler ve teller bir yönsüz çizge (undirected graph) oluşturur, bu çizgede iğneler düğümler (vertices) ve teller ise kenarlardır (edges). Teller dışarıdan görünmemektedir ve onlar hakkında bir şey öğrenmenin tek yolu iğneler üzerinde bir **test cihazı** kullanmaktır: i ve j olmak üzere iki iğne seçersek, $i \neq j$, test cihazı, bu iki iğnenin kutunun içinde doğrudan (directly) veya dolaylı olarak (indirectly) bağlı olup olmadığını söyleyecektir. (Test cihazı çizgede bu iğneler arasında bir yol (path) olup olmadığını söyler.)

Kutunun içindeki bağlantı kümesini oyuncağın **tasarımı** olarak adlandıracağız.

Bu tasarımları sorgulamak ve oluşturmak için özel bir yazılım kullanıyorsunuz. Bu yazılım şu şekilde çalışmakta: Yazılım, oyuncağın "tasarım 0" olarak adlandırdığımız bir tasarımı ile başlar. Bu tasarım için kutunun içindeki bağlantıları size göstermez. Bunun yerine, aşağıdaki 3-aşamalı işlemi tekrar tekrar gerçekleştirebilirsiniz:

1. Bir tasarım numarası a ve birbirine eşit olmayan iki iğne numarası i ve j seçersiniz ($i \neq j$).
2. Yazılım, test cihazını bu iki iğne üzerinde kullanırsak ne olacağını söyler. Başka bir deyişle, i ve j iğnelerinin a tasarımında (doğrudan veya dolaylı olarak) bağlantılı olup olmadığını söyler.
3. Ayrıca, iğneler a tasarımında doğrudan veya dolaylı olarak bağlı değilse, a tasarımındaki tüm bağlantıları ve buna ek olarak i ile j arasında bir doğrudan (direct) bağlantıyı içeren yeni bir tasarım oluşturur. Bu tasarıma bir sonraki kullanılabilir (next available) tasarım numarası verilir. (Dolayısıyla, bu şekilde oluşturulan ilk tasarım 1 numara, sonraki tasarım 2 numara, vb. olarak numaralandırılacaktır.) Bunun a tasarımını değiştirmedikçe, sadece ek bir bağlantıya sahip yeni bir tasarım oluşturduğuna dikkat edin.

Amacınız bu işlemi kullanarak tasarım 0 hakkında mümkün olduğunca çok şey öğrenmektir.

Doğrudan ve dolaylı bağlantıları ayırt etmenin bir yolu olmadığından, tasarım 0 için bağlantı kümesini kesin olarak (exact) belirlemenin her zaman mümkün olmadığına dikkat edin. Örneğin, $n = 3$ olan aşağıdaki iki tasarımı ele alalım:



Test cihazı, her iki tasarım için de seçtiğimiz herhangi bir iğne çiftini bağlı olarak bildirecektir, bu nedenle yukarıda açıklanan yazılımı kullanarak bunları ayırt edemeyiz.

Amacınız, tasarım 0'a eşdeğer herhangi bir tasarımı belirlemektir. Test cihazı tüm iğne çiftleri için her iki tasarımda da aynı sonucu bildirirse, iki tasarım **eşdeğerdir**.

Gerçekleştirme (Implementation)

Bu etkileşimli (interaktif) bir problemdir.

```
void ToyDesign(int n, int max_ops);
```

fonksiyonunu gerçekleştirmelisiniz (implement).

Bu fonksiyon tasarım 0'a eşdeğer bir tasarım belirler. Gerçekleştirmeniz (your implementation) bu hedefe aşağıda açıklanan iki fonksiyonu çağırarak ulaşmalıdır. Çağıracağınız ilk fonksiyon şudur:

```
int Connected(int a, int i, int j);
```

Burada $1 \leq i, j \leq n$, $i \neq j$, $a \geq 0$, ve a şu ana kadar oluşturulmuş tasarımların sayısını geçmemelidir. i ve j iğneleri a tasarımında (doğrudan veya dolaylı olarak) bağlantılıysa, bu fonksiyon a değerini döner. Aksi takdirde, şimdiye kadar oluşturulan tasarımların sayısının bir fazlasını döner; bu, a tasarımının tüm bağlantılarına ve ek olarak i ile j arasındaki direk bağlantıya sahip yeni tasarıma atanan numara olur. `Connected` fonksiyonu en fazla `max_ops` defa çağrılabilir.

Programınız `Connected` işlemlerini bitirdiğinde, tasarım 0'a eşdeğer bir tasarımı tanımlamalıdır. Bir tasarımı tanımlamak için program şu fonksiyonu çağırmalıdır:

```
void DescribeDesign(std::vector<std::pair<int,int>> result);
```

`result` parametresi, iğneler arasındaki doğrudan (direct) bağlantıları temsil eden tamsayı çiftlerinden oluşan bir vektördür. Her tamsayı çifti bir bağlantıya karşılık gelir ve bağlanan iki

İğnenin numarasını içermelidir. Her bir (sırasız) iğne çifti arasında en fazla bir doğrudan (direct) bağlantı olmalıdır. Bir iğne ile kendisi arasında doğrudan bağlantı olmamalıdır. Bu fonksiyonun çağrılması programınızın çalışmasını sonlandırır.

Kısıtlar

- $2 \leq n \leq 200$

Puanlama

- Altgörev 1 (10 puan): $n \leq 200$, $max_ops = 20\ 000$
- Altgörev 2 (20 puan): $n \leq 8$, $max_ops = 20$
- Altgörev 3 (35 puan): $n \leq 200$, $max_ops = 2\ 000$
- Altgörev 4 (35 puan): $n \leq 200$, $max_ops = 1\ 350$

Örnek Etkileşim

Contestant action	Grader action	Explanation
	<code>ToyDesign(4, 20)</code>	Oyuncakta 4 adet iğne bulunmaktadır. <code>Connected</code> 'i en fazla 20 kez çağırarak tasarım 0'a eşdeğer herhangi bir tasarımı belirlemeniz gerekir.
<code>Connected(0, 1, 2)</code>	Returns 1.	İğne 1 ve 2, tasarım 0'da doğrudan (direct) veya dolaylı olarak (indirect) bağlı değildir. Yeni tasarım 1 oluşturulur.
<code>Connected(1, 3, 2)</code>	Returns 2.	İğne 3 ve 2, tasarım 1'de doğrudan (direct) veya dolaylı olarak (indirect) bağlı değildir. Yeni tasarım 2 oluşturulur.
<code>Connected(0, 3, 4)</code>	Returns 0.	İğne 3 ve 4, tasarım 0'da doğrudan (direct) veya dolaylı olarak (indirect) bağlıdır. Yeni tasarım oluşturulmaz.
<code>DescribeDesign({{3, 4}})</code>	-	Sadece bir adet bağlantısı olan bir tasarımı tanımlıyoruz: İğne 3 ve 4.

Örnek Değerlendirici

`ToyDesign.zip` görev ekinde verilen `grader.cpp` örnek değerlendirici, standart girdiden gelen girdiyi aşağıdaki formatta okur:

- İlk satırda iğne sayısı n , doğrudan (direct) bağlantı sayısı m , ve max_ops vardır.
- Sonraki m satırda doğrudan (direct) bağlantılar, iğne çiftleri (tuples of pins) olarak yer alır.

Örnek değerlendirici girdiyi okur ve kullanıcının çözümündeki `ToyDesign` fonksiyonunu çağırır.

Değerlendirici, çözümünüzün çalışmasına bağlı olarak aşağıdaki mesajlardan birini verir:

- "Wrong answer: Number of operations exceeds the limit.", bu mesaj `Connected` fonksiyonuna yapılan çağrıların sayısı max_ops 'u aşarsa verilir.
- "Wrong answer: Wrong design id.", bu mesaj `Connected` fonksiyonu çağırılırken verilen a parametresi, çağrının yapıldığı anda mevcut olmayan bir tasarımın numarasıysa verilir.
- "Wrong answer: Incorrect design.", bu mesaj `DescribeDesign` fonksiyonu ile açıklanan tasarım, tasarım 0'a eşdeğer değilse verilir.
- "OK!", bu mesaj `DescribeDesign` fonksiyonu ile açıklanan tasarım, tasarım 0'a eşdeğerse verilir.

Örnek değerlendiriciyi çözümünüzle beraber derlemek için terminalde aşağıdaki komutu yazabilirsiniz:

```
g++ -std=gnu++11 -O2 -o solution grader.cpp solution.cpp
```

Burada `solution.cpp`, CMS'ye gönderilecek çözüm dosyanızdır. Programı ekte sağlanan örnek girdiyle çalıştırmak için terminalde aşağıdaki komutu yazın:

```
./solution < input.txt
```