



Arkadaş

Kişilerin $0, \dots, n - 1$ ile numaralandığı n kişiden oluşan bir sosyal ağ yapalım. Bu ağda bazı kişi çiftleri arkadaşır. Eğer x kişisi y kişisinin arkadaşı olursa, y kişisi de x kişisinin arkadaşı olur.

Kişiler ağa n aşamada eklenir. Her aşamada bir kişi eklendiğinden aşama sayısı da n 'dir ve bu aşamalar da 0 ile $n - 1$ arasında numaralanmıştır. Her zaman i . kişi i . aşamada ağa eklenir. Aşama 0 'da 0 numaralı kişi ağa eklenen ilk kişidir. Takip eden $n - 1$ aşamanın her birinde sıradaki kişi ağa bir *ev sahibi* (host) tarafından eklenir. Ev sahibi daha önce ağa eklenmiş herhangi bir kişi olabilir. i . aşamada ($0 < i < n$) bu aşamanın ev sahibi yeni gelen i . kişiyi şu üç kuraldan birini kullanarak ağa ekleyebilir:

- *IAmYourFriend* (ben senin arkadaşım): i . kişiyi sadece o anki ev sahibi ile arkadaş yapar.
- *MyFriendsAreYourFriends* (benim arkadaşlarım senin arkadaşların): i . kişiyi o anki ev sahibinin tüm arkadaşları ile arkadaş yapar. Bu kural i . kişiyi o anki ev sahibi ile arkadaş yapmaz.
- *WeAreYourFriends* (biz senin arkadaşımız): i . kişiyi o anki ev sahibi ve onun tüm arkadaşları ile arkadaş yapar.

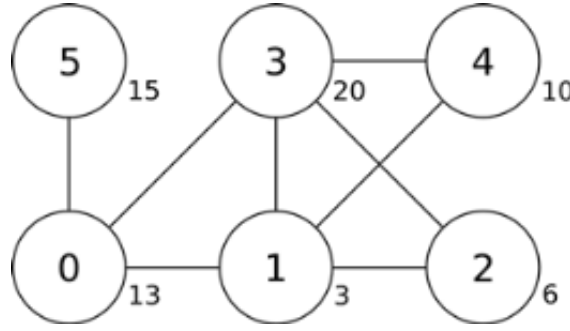
Ağ oluşturduktan sonra, bir anket için bu ağdan bir grup kişiyi örneklem olarak seçmek istiyoruz. Arkadaşlar benzer ilgi alanına sahip olacağından, seçilen grup içinde birbiri ile arkadaş olan hiç kimse olmaması gerekir. Ağdaki her bir kişinin bir *güvenilirlik* (confidence) değeri (bir pozitif tamsayı) ayrıca verilmiştir. Sizden istenen ise toplam güvenilirlik değeri maksimum olan bir örneklem grubu bulmanız.

Örnek

aşama	ev sahibi	kural	eklenen arkadaşlık ilişkisi
1	0	IAmYourFriend	(1, 0)
2	0	MyFriendsAreYourFriends	(2, 1)
3	1	WeAreYourFriends	(3, 1), (3, 0), (3, 2)
4	2	MyFriendsAreYourFriends	(4, 1), (4, 3)
5	0	IAmYourFriend	(5, 0)

En başta ağda yalnızca 0 . kişi vardır. 1 . aşamanın ev sahibi (0 . kişi) ağa 1 . kişiyi *IAmYourFriend* kuralı ile ekler ve böylelikle bu ikisi arkadaş olur. 2 . aşamanın ev sahibi (0 . kişi) ağa 2 . kişiyi *MyFriendsAreYourFriends* kuralı ile ekler, bunun sonucu olarak 2 . kişi ve 1 . kişi (o anki ev sahibi olan 0 . kişinin tek arkadaşı) arkadaş olur. 3 . aşamanın ev sahibi (1 . kişi) ağa 3 . kişiyi *WeAreYourFriends* kuralı ile ekler, bunun sonucu olarak 3 . kişi, o anki ev sahibi olan 1 . kişi ve o anki ev sahibinin arkadaşları olan 0 . kişi ve 2 . kişi ile arkadaş olur. Tabloda 4 . ve 5 . aşamalarda gösterilmiştir. Sonuç ağ ise şekilde verilmiştir. Şekilde kişi numaraları çemberler içinde, kişilerin güvenilirlik değerleri ise

çemberin yanında gösterilmiştir. Örnekte 3. ve 5. kişilerden oluşan örneklem grubunun toplam güvenilirlik değeri $20 + 15 = 35$ olup, olabilecek maksimum toplam güvenilirlik değeridir.



Görev

Her bir aşamanın tanımı ve her bir kişinin güvenilirlik değeri verildiğinde, maksimum toplam güvenilirlik değerli örneklem grubunu bulun. Sizin gerçekleştirmenizi yapacağımız tek fonksiyon `findSample` dir.

- `findSample(n, confidence, host, protocol)`
 - `n`: kişi sayısı.
 - `confidence`: n uzunluğunda bir dizi; `confidence[i]` değeri i . kişinin güvenilirlik değeridir.
 - `host`: n uzunluğunda bir dizi; `host[i]` değeri i . aşamanın ev sahibinin numarasıdır.
 - `protocol`: n uzunluğunda bir dizi; `protocol[i]` değeri i . aşamada ($0 < i < n$) kullanılan kuraldır: 0 değeri `IAmYourFriend`, 1 değeri `MyFriendsAreYourFriends`, ve 2 değeri `WeAreYourFriends` kurallarını belirtir.
 - 0.aşamada ev sahibi olmayacağı için, `host[0]` ve `protocol[0]` değerleri tanımsızdır ve programınız bunları kullanmamalıdır.
 - Fonksiyon olabilecek maksimum toplam güvenilirlik değerini döndürmelidir.

Altgörevler

Bazı altgörevler aşağıda gösterildiği üzere kuralların yalnızca bazılarını kullanır.

altgörev	puan	n	güvenilirlik	kullanılan kural
1	11	$2 \leq n \leq 10$	$1 \leq \text{confidence} \leq 1,000,000$	Tüm kurallar
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	Sadece <code>MyFriendsAreYourFriends</code>
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	Sadece <code>WeAreYourFriends</code>
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	Sadece <code>IAmYourFriend</code>
5	23	$2 \leq n \leq 1,000$	Tüm güvenilirlik değerleri 1	<code>IAmYourFriend</code> ve <code>MyFriendsAreYourFriends</code>
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{confidence} \leq 10,000$	Tüm kurallar

Gerçekleştirim detayları

İsmi `friend.c`, `friend.cpp` ya da `friend.pas` olan tek bir dosya göndermelisiniz. Bu dosya yukarıda tanımlanan fonksiyonu şu formatta gerçekleştirmelidir. C/C++ gerçekleştirmeleri `friend.h` header dosyasını `include` etmelidir.

C/C++ programı

```
int findSample(int n, int confidence[], int host[], int protocol[]);
```

Pascal programı

```
function findSample(n: longint, confidence: array of longint, host: array of longint; protocol: array of longint): longint;
```

Örnek grader

Örnek grader girdiyi aşağıdaki formatta okur:

- satır 1: `n`
- satır 2: `confidence[0], ..., confidence[n-1]`
- satır 3: `host[1], protocol[1], host[2], protocol[2], ..., host[n-1], protocol[n-1]`

Örnek grader `findSample` fonksiyonundan dönen değeri yazar.