



## دوست

می‌خواهیم یک شبکه‌ی اجتماعی شامل  $n$  نفر که از  $0$  تا  $n - 1$  شماره‌گذاری شده‌اند ایجاد کنیم. برخی از افراد این شبکه با یکدیگر دوست هستند. اگر شخص  $x$  با شخص  $y$  دوست باشد، شخص  $y$  نیز با  $x$  دوست خواهد بود.

افراد در  $n$  مرحله به شبکه اضافه می‌شوند، که این مراحل نیز به ترتیب از  $0$  تا  $n - 1$  شماره‌گذاری شده‌اند. شخص  $i$  در مرحله  $i$  اضافه می‌شود. در مرحله  $0$ ، شخص  $0$  به عنوان تنها فرد شبکه اضافه می‌شود. در هر یک از  $n - 1$  مرحله‌ی بعدی، یک فرد توسط یک میزبان به شبکه اضافه می‌شود، که این میزبان می‌تواند هر شخص از اعضای فعلی شبکه باشد. در مرحله  $i$  ( $0 < i < n$ )، میزبان آن مرحله می‌تواند شخص تازه‌وارد  $i$  را با یکی از سه پروتکل زیر به شبکه اضافه کند:

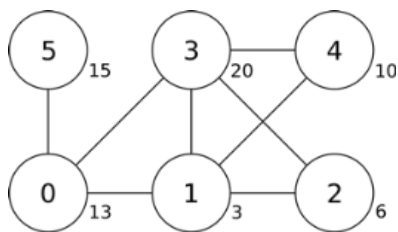
- **IAmYourFriend**: شخص  $i$  را با شخص میزبان دوست می‌کند.
- **MyFriendsAreYourFriends**: شخص  $i$  را با تمام اشخاصی که در آن لحظه با میزبان دوست‌اند، دوست می‌کند. توجه کنید که این پروتکل شخص  $i$  را با خود میزبان دوست نمی‌کند.
- **WeAreYourFriends**: شخص  $i$  را با میزبان و تمام اشخاصی که در آن لحظه با میزبان دوست‌اند، دوست می‌کند.

بعد از ساخت شبکه می‌خواهیم یک نمونه از افراد را برای نظرسنجی انتخاب کنیم، این نمونه یک زیرمجموعه از کل اعضای شبکه است. از آن جایی که دوستان معمولاً علائق مشترکی دارند، در این نمونه نباید هیچ دو فردی که با یکدیگر دوست‌اند، انتخاب شوند. هر فرد یک ضریب اطمینان دارد که به صورت یک عدد صحیح مثبت نشان داده می‌شود. می‌خواهیم نمونه‌ای را بیابیم که بیشترین مجموع ضریب اطمینان را داشته باشد.

## مثال

stage	host	protocol	friend relations added
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)

در ابتدا، شبکه فقط شامل شخص  $0$  است. میزبان مرحله  $1$  (شخص  $0$ ) شخص جدید  $1$  را با پروتکل IAmYourFriend دعوت می‌کند، و در نتیجه این دو شخص با هم دوست می‌شوند. میزبان مرحله  $2$  (مجدداً شخص  $0$ ) شخص  $2$  را با پروتکل MyFriendsAreYourFriends دعوت می‌کند، که باعث می‌شود شخص  $1$  (تنها دوست میزبان) با شخص  $2$  دوست شود. میزبان مرحله  $3$  (شخص  $1$ ) شخص  $3$  را با پروتکل WeAreYourFriends اضافه می‌کند، که باعث می‌شود شخص  $3$  با شخص  $1$  (میزبان) و اشخاص  $0$  و  $2$  (دو دوست میزبان) دوست شود. مراحل  $4$  و  $5$  نیز در جدول بالا نشان داده شده‌اند. شبکه‌ی نهایی در شکل زیر نشان داده شده است. در این شکل، اعداد داخل دایره نشان‌دهنده‌ی شماره‌ی اشخاص، و اعداد کنار دایره نشان‌دهنده‌ی ضریب اطمینان اشخاص است. نمونه‌ی متشکل از اشخاص  $3$  و  $5$  دارای مجموع ضریب اطمینان  $35 = 15 + 20$  است که بیشترین مقدار ممکن در این شبکه است.



## مسئله

با داشتن توصیف تمامی مراحل و ضریب اطمینان اشخاص، نمونه‌ای بیابید که بیش‌ترین مجموع ضریب اطمینان را داشته باشد. شما باید تابع `findSample` را پیاده‌سازی کنید.

`findSample(n, confidence, host, protocol)` •

- `n`: تعداد اشخاص.
- `confidence`: آرایه‌ای به طول `n`; `confidence[i]` نشان‌دهنده‌ی ضریب اطمینان شخص `i` است.
- `host`: آرایه‌ای به طول `n`; `host[i]` نشان‌دهنده‌ی میزبان مرحله‌ی `i` است.
- `protocol`: آرایه‌ای به طول `n`; `protocol[i]` پروتکلی است که در مرحله‌ی `i` ( $0 < i < n$ ) استفاده شده است: 0 برای `IAmYourFriend`، 1 برای `MyFriendsAreYourFriends` و 2 برای `WeAreYourFriends`.
- از آن جایی که مرحله‌ی 0 هیچ میزبانی ندارد، `host[0]` و `protocol[0]` تعریف نشده‌اند و نباید توسط برنامه‌ی شما مورد دست‌رسی قرار گیرند.
- تابع باید بیش‌ترین مجموع ضریب اطمینان ممکن برای نمونه‌ها را برگرداند.

## زیرمسئله‌ها

برخی از زیرمسئله‌ها فقط از زیرمجموعه‌ای از پروتکل‌ها استفاده می‌کنند که در جدول زیر مشخص شده‌اند.

subtask	points	$n$	confidence	protocols used
1	11	$2 \leq n \leq 10$	$1 \leq \text{confidence} \leq 1,000,000$	All three protocols
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	Only MyFriendsAreYourFriends
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	Only WeAreYourFriends
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	Only IAmYourFriend
5	23	$2 \leq n \leq 1,000$	All confidence values are 1	Both MyFriendsAreYourFriends and IAmYourFriend
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{confidence} \leq 10,000$	All three protocols

## جزئیات پیاده‌سازی

شما باید دقیقاً یک فایل با نام `friend.c`، `friend.cpp` یا `friend.pas` ارسال کنید. این فایل باید تابعی که در بالا توضیح داده شد را به شکل زیر پیاده‌سازی کند. همچنین شما باید هدر فایل `friend.h` را در برنامه‌ی C/C++ خود درج کنید.

### برنامه‌ی C/C++

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

### برنامه‌ی پاسکال

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

### ارزیاب نمونه

ارزیاب نمونه ورودی را در قالب زیر می‌خواند:

- خط ۱: `n`.
- خط ۲: `confidence[0]`، ...، `confidence[n-1]`.
- خط ۳: `host[1]`، `protocol[1]`، `host[2]`، `protocol[2]`، ...، `host[n-1]`، `protocol[n-1]`.

ارزیاب نمونه مقدار خروجی تابع `findSample` را چاپ می‌کند.