



## Bạn bè

Chúng ta xây dựng mạng xã hội gồm  $n$  thành viên được đánh số là  $0, \dots, n - 1$ . Có một số cặp thành viên trong mạng sẽ là bạn bè. Nếu thành viên  $x$  trở thành bạn của thành viên  $y$ , thì thành viên  $y$  cũng sẽ là bạn của thành viên  $x$ .

Các thành viên được bổ sung vào mạng theo  $n$  giai đoạn, cũng được đánh số từ  $0$  đến  $n - 1$ . Thành viên  $i$  được bổ sung ở giai đoạn  $i$ . Ở giai đoạn  $0$ , thành viên  $0$  được bổ sung như là thành viên duy nhất của mạng. Trong mỗi giai đoạn thuộc  $n - 1$  giai đoạn tiếp theo, thành viên được bổ sung vào mạng bởi người giới thiệu, là một thành viên bất kỳ của mạng hiện thời. Tại giai đoạn  $i$  ( $0 < i < n$ ), người giới thiệu của giai đoạn này có thể bổ sung thành viên  $i$  vào mạng theo một trong ba cách thức sau:

- *IAmYourFriend* gắn thành viên  $i$  là bạn của duy nhất người giới thiệu.
- *MyFriendsAreYourFriends* gắn thành viên  $i$  là bạn với *bất cứ* người bạn nào của người giới thiệu tại thời điểm này. Chú ý là, cách thức này *không* gắn thành viên  $i$  là bạn của người giới thiệu.
- *WeAreYourFriends* gắn thành viên  $i$  là bạn của người giới thiệu và cũng là bạn của *bất cứ* người bạn nào của người giới thiệu tại thời điểm này.

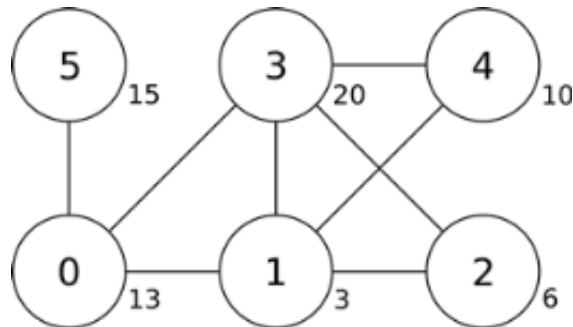
Sau khi xây dựng xong mạng chúng ta muốn xác định một *mẫu thử* để điều tra, nghĩa là cần chọn ra một nhóm thành viên của mạng. Do các người bạn bè thường có những mối quan tâm tương tự nhau, nên mẫu thử phải không được chứa cặp hai thành viên là bạn bè của nhau. Mỗi thành viên có một *độ tự tin*, được đánh giá bởi một số nguyên dương, và chúng ta muốn tìm mẫu thử với tổng độ tự tin là lớn nhất.

## Ví dụ

giai đoạn	người giới thiệu	cách thức	quan hệ bạn bè được bổ sung
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)

Thoạt tiên mạng gồm mỗi người  $0$ . Người giới thiệu ở giai đoạn 1 (thành viên  $0$ ) mời thành viên mới 1 theo cách thức *IAmYourFriend*, vì thế họ trở thành bạn bè của nhau. Người giới thiệu ở giai đoạn 2 (lại vẫn là thành viên  $0$ ) mời thành viên 2 bởi cách thức *MyFriendsAreYourFriends*, theo đó thành viên 1 (là người bạn duy nhất của người giới thiệu) sẽ được gắn là bạn duy nhất của thành viên 2. Người giới thiệu ở giai đoạn 3 (thành viên 1) bổ sung thành viên 3 bởi cách thức *WeAreYourFriends*, theo đó gắn thành viên 3 là bạn của thành viên 1 (là người giới thiệu) và các thành viên  $0$  và  $2$  (là các người

bạn của người giới thiệu). Kết quả của giai đoạn 4 và 5 cũng được mô tả trong bảng ở trên. Mạng kết quả cuối cùng được chỉ ra trong hình vẽ sau đây, trong đó các số trong vòng tròn là chỉ số của thành viên còn các số bên cạnh các vòng tròn là độ tự tin. Mẫu thử gồm hai thành viên 3 và 5 có tổng độ tự tin bằng  $20 + 15 = 35$ , là mẫu có tổng độ tự tin lớn nhất có thể.



## Nhiệm vụ

Cho biết mô tả của mỗi giai đoạn và độ tự tin của mỗi thành viên, hãy xác định mẫu thử với tổng độ tự tin lớn nhất. Bạn chỉ phải cài đặt hàm `findSample`.

- `findSample(n, confidence, host, protocol)`
  - `n`: số lượng thành viên.
  - `confidence`: mảng độ dài `n`; `confidence[i]` chứa độ tự tin của thành viên `i`.
  - `host`: mảng độ dài `n`; `host[i]` chứa chỉ số của người giới thiệu ở giai đoạn `i`.
  - `protocol`: mảng độ dài `n`; `protocol[i]` chứa cách thức được sử dụng ở giai đoạn `i` ( $0 < i < n$ ): 0 là `IAMYourFriend`, 1 là `MyFriendsAreYourFriends`, và 2 là `WeAreYourFriends`.
  - Do không có người giới thiệu ở giai đoạn 0, `host[0]` và `protocol[0]` là không xác định và chương trình của bạn không được truy nhập đến chúng.
  - Hàm phải trả lại tổng độ tự tin lớn nhất của mẫu thử.

## Subtasks

Một số subtasks chỉ sử dụng một bộ phận các cách thức có thể, như được mô tả trong bảng sau đây.

subtask	điểm	$n$	độ tự tin	các cách thức được dùng
1	11	$2 \leq n \leq 10$	$1 \leq \text{confidence} \leq 1,000,000$	tất cả các cách thức
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	chỉ dùng <code>MyFriendsAreYourFriends</code>
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	chỉ dùng <code>WeAreYourFriends</code>
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	chỉ dùng <code>IAMYourFriend</code>
5	23	$2 \leq n \leq 1,000$	Tất cả các độ tự tin đều là 1	Dùng <code>MyFriendsAreYourFriends</code> và <code>IAMYourFriend</code>
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{confidence} \leq 10,000$	tất cả các cách thức

## Chi tiết cài đặt

Bạn cần nộp một file duy nhất đặt tên là `friend.c`, `friend.cpp` hay `friend.pas`. File này cài đặt chương trình con mô tả ở trên sử dụng các đặc tả sau đây. Bạn cần nạp header file `friend.h` đối với cài đặt trên C/C++.

### C/C++ program

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

### Pascal programs

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

### Sample grader

Sample grader đọc dữ liệu vào theo khuôn dạng:

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

Sample grader sẽ in ra giá trị được trả về bởi `findSample`.