



친구

$0, \dots, n - 1$ 로 번호가 매겨진 n 명의 사람으로 구성된 소셜 네트워크를 만들자. 이 네트워크에 있는 사람들 중 어떤 쌍은 서로 친구가 된다. 즉, 사람 x 가 사람 y 의 친구가 되면, 사람 y 도 사람 x 의 친구가 된다.

사람들은 n 단계를 거쳐서 네트워크에 추가되는데, 각 단계를 0 부터 $n - 1$ 로 번호를 매기자. 사람 i 는 단계 i 에 추가된다. 단계 0 에서, 사람 0 은 이 네트워크에 있는 유일한 사람으로 추가된다. 다음 $n - 1$ 개의 각 단계에서, **초대자**가 새로 사람 하나를 네트워크에 추가한다. 초대자는 이 단계에 네트워크에 이미 들어와 있는 사람 중 하나이다. 단계 i 에서 ($0 < i < n$), 이 단계의 초대자는 새로 들어오는 사람 i 를 다음 프로토콜 중 하나를 사용하여 네트워크에 추가한다.

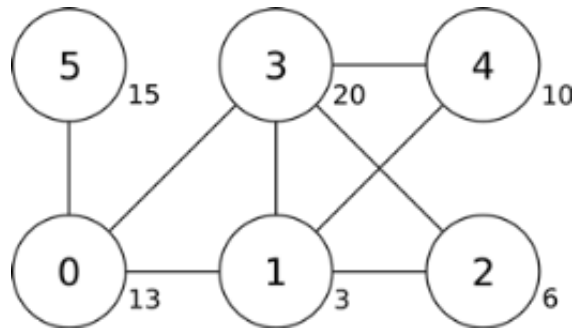
- **IAmYourFriend**는 사람 i 를 초대자의 친구로만 등록한다.
- **MyFriendsAreYourFriends**는 사람 i 를 초대자의 현재 **모든** 친구의 친구로 등록한다. 이 경우 사람 i 는 초대자의 친구가 **아니다**.
- **WeAreYourFriends**는 사람 i 를 초대자와, 초대자의 현재 **모든** 친구의 친구로 등록한다.

네트워크를 다 만든 다음에는 설문을 위해서 **표본**을 구하고자 한다. 표본은 네트워크에 속한 사람들로 구성된 모임이다. 친구끼리는 서로 좋아하는 것이 비슷하기 때문에, 표본에 속하는 사람들 중에는 서로 친구인 쌍이 있으면 안된다. 각 사람마다 설문에서 쓰이는 **신뢰도**가 있는데, 이는 양의 정수로 표현된다. 우리는 신뢰도의 총합이 최대인 표본을 구하려 한다.

예제

단계	초대자	프로토콜	추가되는 친구 관계
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)

처음 네트워크에는 사람 0 만 있다. 단계 1의 초대자(사람 0)는 새로 사람 1 을 IAmYourFriend 프로토콜로 추가해서, 서로 친구가 된다. 단계 2의 초대자(이번에도 사람 0)는 사람 2 를 MyFriendsAreYourFriends로 추가하는데, 사람 1 (초대자의 유일한 친구)이 사람 2 의 유일한 친구가 된다. 단계 3의 초대자(사람 1)는 사람 3 을 WeAreYourFriends로 추가하는데, 사람 3 은 사람 1 (초대자)과 사람 $0, 2$ (초대자의 친구들)의 친구가 된다. 단계 4와 단계 5도 표에 보인 바와 같다. 최종적인 네트워크는 다음 그림과 같고, 동그라미 안의 숫자는 사람의 번호이고, 동그라미 아래의 숫자는 이 사람의 신뢰도를 나타낸다. 사람 3 과 5 로 이루어진 표본의 신뢰도의 총합은 $20 + 15 = 35$ 인데, 이는 가능한 신뢰도의 총합 중 최대이다.



문제

각 단계의 기술과 각 사람의 신뢰도가 주어졌을 때, 신뢰도의 총합이 최대인 표본을 구하시오. 다음과 같은 findSample을 구현하면 된다.

- findSample(n , confidence, host, protocol)
 - n : 사람의 수.
 - confidence: 크기 n 인 배열; confidence[i]는 사람 i 의 신뢰도.
 - host: 크기 n 인 배열; host[i]는 단계 i 의 초대자.
 - protocol: 크기 n 인 배열; protocol[i]는 단계 i 에서 사용하는 프로토콜 ($0 < i < n$): 이 값이 0이면 IAmYourFriend, 1이면 MyFriendsAreYourFriends, 2이면 WeAreYourFriends.
 - 단계 0에서는 초대자가 없기 때문에, host[0]와 protocol[0]는 정의되지 않고 프로그램에서 사용하면 안된다.
 - 이 함수의 리턴값은 신뢰도의 총합이 최대인 표본에 대한 신뢰도의 총합

부분 문제

다음 표에서 보인 바와 같이, 어떤 부분 문제에서는 프로토콜 중 일부만 사용한다.

부분문제	점수	n	신뢰도	사용하는 프로토콜
1	11	$2 \leq n \leq 10$	$1 \leq \text{신뢰도} \leq 1,000,000$	세 프로토콜 다 사용
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{신뢰도} \leq 1,000,000$	MyFriendsAreYourFriends만 사용
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{신뢰도} \leq 1,000,000$	WeAreYourFriends만 사용
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{신뢰도} \leq 1,000,000$	IAmYourFriend만 사용
5	23	$2 \leq n \leq 1,000$	모든 신뢰도는 1	MyFriendsAreYourFriends와 IAmYourFriend만 사용
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{신뢰도} \leq 10,000$	세 프로토콜 다 사용

구현 주의 사항

정확히 하나의 파일을 제출해야 하며, 이름은 friend.c, friend.cpp 또는 friend.pas이다. 이 파일에 위에서 언급한 서버프로그램을 다음과 같은 선언을 사용하여 구현해야 한다. C/C++의 경우 헤더 파일 friend.h을 #include해야 한다.

C/C++ 프로그램

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

Pascal 프로그램

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

Sample grader

Sample grader는 다음 형식으로 된 입력을 읽는다.

- 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 grader는 findSample의 리턴 값을 출력할 것이다.