



## Приятел

Социална мрежа ще обединява  $n$  потребители, номерирани от 0 до  $n - 1$ . Някои двойки потребители на мрежата ще станат приятели. Ако потребителят  $x$  стане приятел с потребителя  $y$ , то потребителят  $y$  също ще бъде приятел с потребителя  $x$ .

Потребителите ще бъдат включени в мрежата в  $n$  стъпки, също номерирани от 0 до  $n - 1$ . Потребителят  $i$  ще бъде включен в мрежата на  $i$ -тата стъпка. На нулевата стъпка потребителят 0 ще бъде включен като единствен потребител на мрежата. На всяка от следващите  $n - 1$  стъпки, поредният потребител бива включен от мрежата от някой от вече включените в мрежата потребители, наричан *домакин* на стъпката. На стъпка  $i$  ( $0 < i < n$ ) домакинът на стъпката може да добави потребителя  $i$ , следвайки един от възможните три протокола:

- *IamYourFriend* включва потребителя  $i$  като приятел само на домакина.
- *MyFriendsAreYourFriends* включва потребителя  $i$  като приятел на всички приятели на домакина. Забележете, че този протокол не прави потребителя  $i$  приятел на домакина.
- *WeAreYourFriends* включва потребителя  $i$  като приятел на домакина и на всички приятели на домакина.

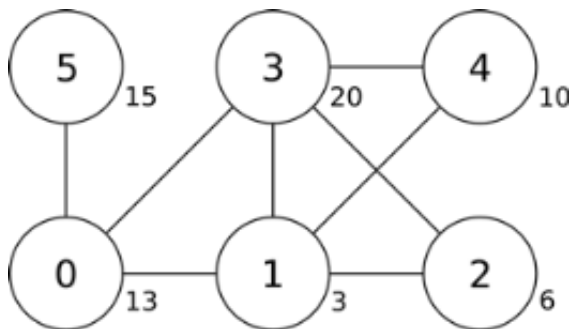
След създаването на мрежата, бихме искали да направим *извадка* за някакво изследване, т.е. да изберем група от потребители на мрежата за участие в изследването. Тъй като приятелите обикновено имат сходни интереси, в групата не бива да има двойки потребители, които са приятели. С всеки потребител е свързано положително цяло число, което наричаме *значимост* на потребителя за изследването. Бихме искали да изберем групата така, че сборът на значимостите на избраните потребители да е максимален.

## Пример

стъпка	домакин	протокол	добавени двойки
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)

В началото в групата е само потребителят 0. Бидейки домакин на стъпка 1, той включва потребителя 1 с протокола *IamYourFriend* и двамата стават приятели. Домакин на стъпка 2 е отново потребителят 0 и той добавя потребител 2 с протокола *MyFriendsAreYourFriends*. Така потребителят 1, който е единствен приятел на 0, става единствен приятел на 2. Домакинът на стъпка три е 1 и той добавя потребител 3 с протокола *WeAreYourFriends* при което потребител

три става приятел на домакина 1 и на неговите двама приятели 0 и 2. Стъпки 4 и 5, показани също в таблицата по-горе, довеждат мрежата до вида показан на Фигурата, където числата в кръгчетата са номерата на потребителите, а числото в близост до всяко кръгче задава значимостта на съответния потребител за изследването. Извадката от потребител 3 и потребител 5 е с обща значимост от  $20 + 15 = 35$  и това е извадката с максимална значимост.



## Task

По зададени описание на стъпките и значимостите на потребителите, трябва да се намери допустимата извадка с максимална значимост. Напишете функция `findSample`:

- `findSample(n, confidence, host, protocol)`
  - `n`: брой на потребителите.
  - `confidence`: масив с  $n$  елемента; `confidence[i]` съдържа значимостта на  $i$ -тия потребител.
  - `host`: масив с  $n$  елемента; `host[i]` съдържа домакина на  $i$ -тата стъпка.
  - `protocol`: масив с  $n$  елемента; в `protocol[i]` е зададен номерът на протокола използван при  $i$ -тата стъпка,  $0 < i < n$ : 0 за `IamYourFriend`, 1 за `MyFriendsAreYourFriends` или 2 за `WeAreYourFriends`.
  - Тъй като на стъпка 0 няма домакин и протокол, стойностите `host[0]` и `protocol[0]` са неопределени и не бива да се използват.
  - Функцията трябва да връща максималната значимост на допустима извадка.

## Подзадачи

Някои подзадачи използват подмножество от протоколи, както е показано в таблицата по-долу.

подзад.	точки	$n$	значимост	протоколи
1	11	$2 \leq n \leq 10$	$1 \leq \text{значимост} \leq 1000000$	Всеки от трите
2	8	$2 \leq n \leq 1000$	$1 \leq \text{значимост} \leq 1000000$	Само <code>MyFriendsAreYourFriends</code>
3	8	$2 \leq n \leq 1000$	$1 \leq \text{значимост} \leq 1000000$	Само <code>WeAreYourFriends</code>
4	19	$2 \leq n \leq 1000$	$1 \leq \text{значимост} \leq 1000000$	Само <code>IamYourFriend</code>
5	23	$2 \leq n \leq 1000$	Всички значимости са 1	<code>MyFriendsAreYourFriends</code> и <code>IamYourFriend</code>
6	31	$2 \leq n \leq 100000$	$1 \leq \text{значимост} \leq 10000$	Всеки от трите

## Детайли на имплементацията за C/C++

Трябва да изпратите само един файл с име `friend.c` или `friend.cpp`. Той трябва да съдържа имплементация на функцията `findSample` такава, каквато е описана по-горе, със следната спецификация:

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

като не забравите да включите файла `friends.h`.

### Примерен грейдър

Примерният грейдър чете вход във формат:

- 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]`

Примерният грейдър извежда резултата от извикването на функцията `findSample`.