



Ystävä

Rakennamme sosiaalisen verkoston n henkilöstä, jotka on numeroitu $0, \dots, n - 1$. Osa verkoston henkilöistä tulee olemaan ystäviä. Jos henkilö x on henkilön y ystävä, myös henkilö y on henkilön x ystävä.

Henkilöt lisätään verkostoon n vaiheessa, jotka on myös numeroitu $0, \dots, n - 1$. Henkilö i lisätään vaiheessa i . Vaiheessa 0 henkilö 0 lisätään ainoana henkilönä verkostoon. Kaikissa seuraavissa $n - 1$ vaiheissa uuden henkilön lisää verkostoon *kutsuja*, joka voi olla kuka tahansa henkilö, joka on jo verkostossa. Vaiheessa i ($0 < i < n$) tämän vaiheen kutsuja voi lisätä uuden henkilön i verkostoon jollakin seuraavista tavoista:

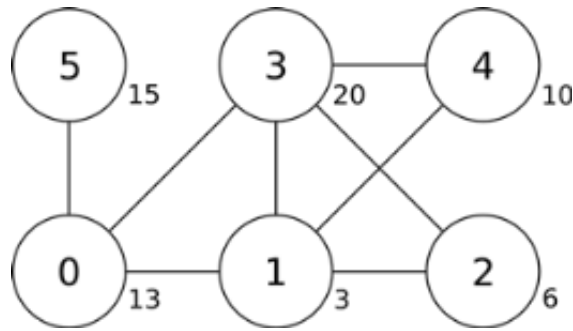
- *IAmYourFriend* tekee henkilöstä i vain kutsujan ystävän.
- *MyFriendsAreYourFriends* tekee henkilöstä i *kaikkien* kutsujan senhetkisten ystävien ystävän. Huomaa, että henkilöstä i *ei* tule kutsujan ystävää.
- *WeAreYourFriends* tekee henkilöstä i kutsujan ystävän sekä lisäksi *kaikkien* kutsujan senhetkisten ystävien ystävän.

Verkoston rakentamisen jälkeen haluamme valita *otoksen* tutkimusta varten eli valita ryhmän verkoston henkilöistä. Koska ystävillä on usein samankaltaisia kiinnostuksenkohteita, otoksessa ei saa olla ketään kahta henkilöä, jotka ovat keskenään ystäviä. Jokaisesta henkilöstä on tiedossa *luotettavuusarvo*, positiivinen kokonaisluku, ja haluamme valita otoksen, jonka kokonaisluotettavuus eli luotettavuusarvojen summa on suurin mahdollinen.

Esimerkki

vaihe	kutsuja	lisäystapa	uudet ystävydet
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)

Alluksi verkostossa on vain henkilö 0. Vaiheen 1 kutsuja (henkilö 0) lisää uuden henkilön 1 tavalla IAmYourFriend, ja heistä tulee ystäviä. Vaiheen 2 kutsuja (henkilö 0 jälleen) lisää henkilön 2 tavalla MyFriendsAreYourFriends, jolloin henkilöstä 1 (kutsujan ainoa ystävä) tulee henkilön 2 ainoa ystävä. Vaiheen 3 kutsuja (henkilö 1) lisää henkilön 3 tavalla WeAreYourFriends, jolloin henkilöstä 3 tulee henkilön 1 (kutsuja) sekä henkilöiden 0 ja 2 (kutsujan ystävät) ystävä. Vaiheet 4 ja 5 on myös näytetty yllä olevassa taulukossa. Lopullinen verkosto on näytetty seuraavassa kuvassa, jossa ympyröidyt numerot ovat henkilöiden tunnuksot ja ympyröiden viereiset luvut ovat luotettavuusarvot. Henkilöistä 3 ja 5 muodostuvan otoksen kokonaisluotettavuus on $20 + 15 = 35$, joka on suurin mahdollinen.



Tehtävä

Annettuna on jokaisen vaiheen kuvaus sekä kunkin henkilön luotettavuusarvo. Tehtäväsi on etsiä otos, jonka kokonaisluotettavuus on suurin. Sinun täytyy toteuttaa funktio `findSample`.

- `findSample(n, confidence, host, protocol)`
 - `n`: henkilöiden lukumäärä.
 - `confidence`: taulukko kokoa `n`; `confidence[i]` on henkilön `i` luotettavuusarvo.
 - `host`: taulukko kokoa `n`; `host[i]` on vaiheen `i` kutsuja.
 - `protocol`: taulukko kokoa `n`; `protocol[i]` on vaiheessa `i` ($0 < i < n$) käytettävä lisäystapa: 0 tavalle `IAmYourFriend`, 1 tavalle `MyFriendsAreYourFriends` sekä 2 tavalle `WeAreYourFriends`.
 - Koska vaiheessa 0 ei ole kutsujaa, `host[0]` ja `protocol[0]` ovat määrittelemättömiä ja niitä ei tule käyttää ohjelmassa.
 - Funktion tulee palauttaa suurin mahdollinen otoksen kokonaisluotettavuus.

Osatehtävät

Joissakin osatehtävissä on käytössä vain lisäystapojen osajoukko seuraavan taulukon mukaisesti.

osatehtävä	pisteet	n	luotettavuus (c)	lisäystavat
1	11	$2 \leq n \leq 10$	$1 \leq c \leq 1\,000\,000$	Kaikki lisäystavat
2	8	$2 \leq n \leq 1\,000$	$1 \leq c \leq 1\,000\,000$	Vain <code>MyFriendsAreYourFriends</code>
3	8	$2 \leq n \leq 1\,000$	$1 \leq c \leq 1\,000\,000$	Vain <code>WeAreYourFriends</code>
4	19	$2 \leq n \leq 1\,000$	$1 \leq c \leq 1\,000\,000$	Vain <code>IAmYourFriend</code>
5	23	$2 \leq n \leq 1\,000$	Luotettavuus on aina 1.	<code>MyFriendsAreYourFriends</code> ja <code>IAmYourFriend</code>
6	31	$2 \leq n \leq 100\,000$	$1 \leq c \leq 10\,000$	Kaikki lisäystavat

Toteutus

Sinun tulee lähettää tarkalleen yksi tiedosto nimeltä `friend.c`, `friend.cpp` tai `friend.pas`. Tiedoston tulee toteuttaa yllä kuvattu funktio käyttäen seuraavia runkoja. Sinun täytyy myös liittää

mukaan otsikkotiedosto friend.h C/C++-toteutuksessa.

C/C++-ohjelma

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

Pascal-ohjelma

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

Esimerkkitarkastin

Esimerkkitarkastin lukee syötteen seuraavassa muodossa:

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

Esimerkkitarkastin tulostaa funktion findSample palautusarvon.