



Friend

Pravimo novu socijalnu mrežu koja će se sastojati od n osoba numerisanih brojevima $0, \dots, n - 1$. Među ovim osobama, neki parovi će biti prijatelji. Ako osoba x postane prijatelj osobe y , tada osoba y također postaje prijatelj osobe x .

Osobe se dodaju u mrežu u n faza, koje su također numerisane brojevima od 0 do $n - 1$. Osoba i se dodaje u fazi i . U fazi 0 , osoba 0 se dodaje kao jedina osoba u mreži. U svakoj od narednih $n - 1$ faza, osoba se dodaje u mrežu od strane *domaćina*, koji može biti bilo koja osoba koja je već u mreži u tom trenutku. U fazi i ($0 < i < n$), domaćin te faze može da doda dolazeću osobu i u mrežu koristeći jedan od sljedeća tri protokola:

- *IAmYourFriend* čini da osoba i postane prijatelj samo sa domaćinom.
- *MyFriendsAreYourFriends* čini da osoba i postane prijatelj sa *svakom* osobom koja je prijatelj domaćina u tom trenutku. Obratite pažnju da ovaj protokol *ne* čini da osoba i postane prijatelj domaćina.
- *WeAreYourFriends* čini da osoba i postane prijatelj domaćina, kao i sa *svakom* osobom koja je prijatelj domaćina u tom trenutku.

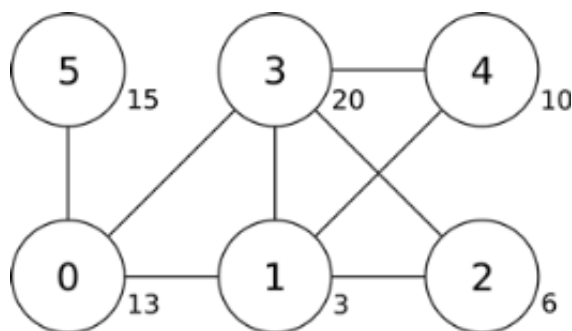
Nakon što se mreža napravi mi želimo da izaberemo *uzorak* za anketu, tj. da izaberemo grupu osoba iz mreže. Kako prijatelji obično imaju slična interesovanja, uzorak ne smije sadržati nijedan par osoba koji su međusobno prijatelji. Svaka osoba ima pouzdanost pri anketiranju koji je neki pozitivan cijeli broj. Mi bismo željeli da nađemo uzorak sa maksimalnom pouzdanošću; pouzdanost uzorka je zbir pouzdanosti osoba u njemu.

Primer

| faza | domaćin | protokol | dodate relacije prijateljstva |
|------|---------|-------------------------|-------------------------------|
| 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) |

Na početku se mreža sastoji samo od osobe 0 . Domaćin faze 1 (osoba 0) dodaje novu osobu 1 koristeći *IAmYourFriend* protokol pa oni postaju prijatelji. Domaćin faze 2 (opet osoba 0) dodaje osobu 2 protokolom *MyFriendsAreYourFriends*, koji čini osobu 1 (jedinim prijateljem domaćina) jedinim prijateljem osobe 2 . Domaćin faze 3 (osoba 1) dodaje osobu 3 koristeći *WeAreYourFriends* protokol, koji čini osobu 3 prijateljem osobe 1 (domaćin) i prijateljima osoba 0 i 2 (domaćinovi prijatelji). Faze 4 i 5 su također prikazane u tabeli iznad. Konačni izgled mreže je prikazan na sljedećoj slici, na kojoj brojevi unutar krugova predstavljaju redne brojeve osoba a brojevi pored krugova predstavljaju pouzdanosti

odgovarajućih osoba. Uzorak koji se sastoji od osoba 3 i 5 ima pouzdanost $20 + 15 = 35$, što je najveća moguća pouzdanost.



Zadatak

Za dati opis svake faze i vrijednosti kvaliteta svake osobe, nađite najveću moguću pouzdanost uzorka. Potrebno je implementirati funkciju `findSample`.

- `findSample(n, confidence, host, protocol)`
 - `n`: broj osoba.
 - `confidence`: niz dužine `n`; `confidence[i]` predstavlja pouzdanost osobe `i`.
 - `host`: niz dužine `n`; `host[i]` predstavlja domaćina faze `i`.
 - `protocol`: niz dužine `n`; `protocol[i]` predstavlja protokol koji je korišćen u fazi `i` ($0 < i < n$): 0 za `IAMYourFriend`, 1 za `MyFriendsAreYourFriends`, i 2 for `WeAreYourFriends`.
 - Kako ne postoji domaćin u fazi 0, `host[0]` i `protocol[0]` su nedefinisani i vaš program ne bi trebalo da pristupa ovim lokacijama.
 - Funkcija treba da vrati najveću moguću vrijednost uzorka.

Podzadaci

Neki podzadaci koriste samo podskup mogućih protokola, kao što je prikazano u sljedećoj tabeli.

| podzadatak | poeni | n | confidence | korišćeni protokoli |
|------------|-------|-------------------------|---|---|
| 1 | 11 | $2 \leq n \leq 10$ | $1 \leq \text{confidence} \leq 1,000,000$ | Sva tri protokola |
| 2 | 8 | $2 \leq n \leq 1,000$ | $1 \leq \text{confidence} \leq 1,000,000$ | Samo <code>MyFriendsAreYourFriends</code> |
| 3 | 8 | $2 \leq n \leq 1,000$ | $1 \leq \text{confidence} \leq 1,000,000$ | Samo <code>WeAreYourFriends</code> |
| 4 | 19 | $2 \leq n \leq 1,000$ | $1 \leq \text{confidence} \leq 1,000,000$ | Samo <code>IAMYourFriend</code> |
| 5 | 23 | $2 \leq n \leq 1,000$ | Sve vrednosti u confidence nizu su 1 | <code>MyFriendsAreYourFriends</code> i <code>IAMYourFriend</code> |
| 6 | 31 | $2 \leq n \leq 100,000$ | $1 \leq \text{confidence} \leq 10,000$ | Sva tri protokola |

Detalji implementacije

Potrebno je da pošaljete tačno jedan fajl koji je potrebno nazvati `friend.c`, `friend.cpp` ili `friend.pas`. Ovaj fajl treba da implementira gore opisanu funkciju koristeći sljedeće potpise. Takođe je potrebno include-ovati i header fajl `friend.h` za C/C++ implementaciju.

C/C++ programi

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

Pascal programi

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

Opis grejdera

Grejder koji se nalazi na vašem računaru čita ulazne podatke u sledećem obliku:

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

Ovaj grejder će štampati vrednost koju vraća funkcija `findSample`.