



Friend

We bouwen een sociaal netwerk van n personen genummerd $0, \dots, n - 1$. Sommige paren van mensen in het netwerk worden vrienden. Als persoon x een vriend wordt van persoon y , dan wordt persoon y ook een vriend van persoon x .

De mensen worden aan het netwerk toegevoegd in n fases, die ook genummerd zijn van 0 to $n - 1$. Persoon i wordt toegevoegd in fase i . In fase 0 wordt persoon 0 toegevoegd als de enige persoon van het netwerk. In elk van de volgende $n - 1$ fases, wordt een persoon toegevoegd aan het netwerk door een *gastheer* (host): dat kan eender welke persoon zijn die al in het netwerk zit. In fase i ($0 < i < n$), kan de gastheer voor die fase de inkomende persoon i toevoegen aan het netwerk volgens een van de drie volgende protocols:

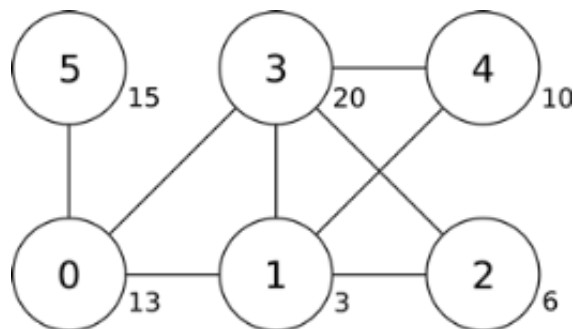
- *IAmYourFriend* maakt persoon i enkel bevriend met de gastheer.
- *MyFriendsAreYourFriends* maakt persoon i bevriend met *elke* persoon die een vriend is van de gastheer op dat moment. Maar let op: dit protocol maakt persoon i *niet* bevriend met de gastheer zelf.
- *WeAreYourFriends* maakt person i bevriend met de gastheer, en met *elke* persoon die een vriend is van de gastheer op dat moment.

Na het bouwen van het netwerk willen we een *steekproef* (sample) kiezen voor een enquête, d.w.z. een groep mensen selecteren uit dit netwerk. Gezien vrienden meestal gelijkaardige interesses hebben, mag de steekproef geen paar personen bevatten die met elkaar bevriend zijn. Elke persoon heeft een *betrouwbaarheid* (confidence), uitgedrukt als een positieve integer, en we willen de steekproef selecteren met een maximale totale betrouwbaarheidsscore.

Voorbeeld

fase	gastheer	protocol	vriend-relaties toegevoegd
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)

In het begin bevat het netwerk enkel persoon 0. De gastheer in fase 1 (persoon 0) nodigt persoon 1 uit met het protocol IAmYourFriend, dus zij worden vrienden. De gastheer in fase 2 (opnieuw persoon 0) nodigt persoon 2 uit via MyFriendsAreYourFriends, en zo wordt persoon 1 (de enige vriend van de gastheer) ook de enige vriend van persoon 2. De gastheer in fase 3 (persoon 1) voegt persoon 3 toe via WeAreYourFriends, en zo wordt persoon 3 bevriend met persoon 1 (de gastheer) en personen 0 en 2 (de vrienden van de gastheer). Fases 4 en 5 worden ook beschreven in de vorige tabel. Het uiteindelijke netwerk wordt getoond in de volgende figuur. De nummers in cirkels zijn de nummers van de personen zelf, en de nummers naast de cirkels zijn hun betrouwbaarheidsscores. De steekproef die bestaat uit personen 3 en 5 heeft een totale betrouwbaarheid van $20 + 15 = 35$, wat de maximale totale betrouwbaarheid is.



Taak

Gegeven de beschrijving van elke fase en de betrouwbaarheidsscore voor elke persoon, moet je een steekproef vinden met een maximale totale betrouwbaarheid. Je moet enkel de functie `findSample` implementeren.

- `findSample(n, confidence, host, protocol)`
 - `n`: het aantal personen.
 - `confidence`: array van lengte `n`; `confidence[i]` is de betrouwbaarheidsscore van persoon `i`.
 - `host`: array van lengte `n`; `host[i]` is de gastheer in fase `i`.
 - `protocol`: array van lengte `n`; `protocol[i]` is het protocol gebruikt in fase `i` ($0 < i < n$): 0 voor IAmYourFriend, 1 voor MyFriendsAreYourFriends, en 2 voor WeAreYourFriends.
 - Gezien er geen gastheer is in fase 0, zijn `host[0]` en `protocol[0]` niet gedefinieerd, en zou jouw programma deze waarden niet mogen gebruiken.
 - De functie moet de maximale totale betrouwbaarheid van een steekproef teruggeven.

Subtaken

Sommige subtaken gebruiken enkel een subset van protocols, zie de volgende tabel:

subtaak	ptn	n	betrouwbaarheidsscores	gebruikte protocols
1	11	$2 \leq n \leq 10$	$1 \leq \text{betrouwbaarheid} \leq 1,000,000$	Alle drie protocols
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{betrouwbaarheid} \leq 1,000,000$	Enkel MyFriendsAreYourFriends
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{betrouwbaarheid} \leq 1,000,000$	Enkel WeAreYourFriends
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{betrouwbaarheid} \leq 1,000,000$	Enkel IAmYourFriend
5	23	$2 \leq n \leq 1,000$	Alle betrouwbaarheden zijn 1	MyFriendsAreYourFriends en IAmYourFriend
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{betrouwbaarheid} \leq 10,000$	Alle drie protocols

Implementatiedetails

Je moet exact één bestand indienen, genaamd `friend.c`, `friend.cpp` of `friend.pas`. Dit bestand moet de hierboven beschreven subroutine implementeren. Bij C/C++ implementaties moet je ook een header-file `friend.h` "includeren".

C/C++ programma

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

Pascal programma

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

Voorbeeldgrader

De voorbeeldgrader leest input in het volgende formaat:

- lijn 1: n
- lijn 2: $\text{confidence}[0], \dots, \text{confidence}[n-1]$
- lijn 3: $\text{host}[1], \text{protocol}[1], \text{host}[2], \text{protocol}[2], \dots, \text{host}[n-1], \text{protocol}[n-1]$

De voorbeeldgrader zal de return-waarde van `findSample` weergeven.