



Priateľ (Friend)

Ideme si spraviť sociálnu sieť. Bude v nej konkrétnych n ľudí. Títo ľudia majú čísla od 0 do $n - 1$. Niektoré dvojice ľudí sa budú priateliť. Priateľstvo je symetrické: ak sa x stane priateľom y , tak aj y sa stane priateľom x .

Našich n ľudí budeme pridávať do našej sociálnej siete postupne, v n krokoch. Kroky si tiež očísľujeme od 0 do $n - 1$. Pre každé i bude platiť, že v kroku i do siete pridáme človeka i . V kroku 0 teda do prázdnej siete pridáme človeka 0. Následne pre každé $i > 0$ pridávaného človeka i musí do siete pozvať niekto, kto už v sieti je - teda konkrétna osoba s číslom menším ako i .

Toho, kto do siete pozýva nového člena, budeme volať hostiteľ (v anglickom zadaní "host"). Áno, znie to v slovenčine debilne, ale jeden zo vstupných parametrov sa volá `host` a použitie iného prekladu by ťa zbytočne poplietlo. Tak to prosím nejak vydrž.

Hostiteľ môže pri pozývaní nového člena (s číslom i) použiť jeden z troch typov pozvánky (protocol):

- *ČauKamoš*: Človek i sa stane priateľom len hostiteľa (a nikoho iného spomedzi ľudí, ktorí už sú v sieti).
- *MámeRovnakýchKamošov*: Človek i sa stane priateľom každého človeka, ktorý sa v tomto okamihu priateli s hostiteľom. (*Nestane* sa ale priateľom samotného hostiteľa.)
- *ČauKamošMámeRovnakýchKamošov*: Oba vyššie spôsoby dokopy. Teda človek i sa stane aj priateľom svojho hostiteľa, aj priateľom všetkých ľudí, ktorí sa s hostiteľom v danom okamihu priatelia.

Po tom, ako zostrojíme celú sociálnu sieť, by sme chceli vybrať skupinu ľudí, ktorých pôjdeme otravovať anketou. Aby sa ľudia vyplňajúci anketu navzájom neovplyvňovali, nesmú sa žiadni dvaja z ľudí, ktorým anketu nanútime, priateliť.

Každý človek má nejakú úroveň dôvery v svoje schopnosti (sebadôvera, po anglicky confidence). Táto úroveň je vyjadrená kladným celým číslom. Radi by sme dostali na našu anketu odpovede od ľudí, ktorí si dôverujú. Presnejšie, chceme nájsť vyhovujúcu skupinu ľudí, ktorá bude mať najväčší možný súčet úrovní sebadôvery.

Príklad

Nasleduje príklad popisujúci ako postupne tvoríme sociálnu sieť s $n = 6$ ľuďmi. Na začiatku je sieť prázdna. Potom do nej v kroku 0 pridáme človeka 0. Ten zatiaľ nemá žiadnych priateľov. (Tento krok sme v tabuľke ani neuviedli.)

V kroku 1 hostiteľ pre tento krok (človek 0) pridá nového člena (človeka 1). Použije na to pozvánku typu `SomTvojKamoš`, takže odteraz sú ľudia 0 a 1 priateľmi.

V kroku 2 hostiteľ pre tento krok (opäť človek 0) pridá nového člena (človeka 2). Tentokrát na to ale použije pozvánku typu `MámeRovnakýchKamošov`. V tejto chvíli je priateľom hostiteľa (človeka 0)

jedine človek 1. Ten sa teda stane priateľom človeka 2. Pribudlo teda práve jedno nové priateľstvo: medzi ľuďmi 1 a 2.

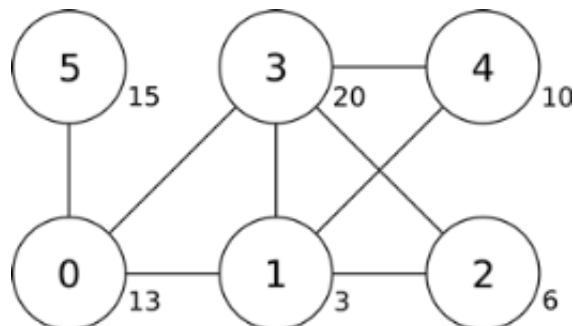
V kroku 3 hostiteľ pre tento krok (tentokrát človek 1) pridá človeka 3 pozvánkou typu ČauKamošMámeRovnakýchKamošov. Hostiteľ má v súčasnosti dvoch priateľov: ľudí 0 a 2. Pribudnú teda tri priateľstvá: medzi človekom 3 a každým z ľudí 0, 1 a 2.

Kroky 4 a 5 podobne pridajú ľudí 4 a 5. Nájdete ich v nižšie uvedenej tabuľke.

krok	hostiteľ	typ pozvánky	priateľstvá, ktoré pribudnú
1	0	ČauKamoš	(1, 0)
2	0	MámeRovnakýchKamošov	(2, 1)
3	1	ČauKamošMámeRovnakýchKamošov	(3, 1), (3, 0), (3, 2)
4	2	MámeRovnakýchKamošov	(4, 1), (4, 3)
5	0	ČauKamoš	(5, 0)

Výsledná sociálna sieť je znázornená na nasledujúcom obrázku. Čísla vo vnútri krúžkov predstavujú ľudí, čiary medzi krúžkami predstavujú priateľstvá. Čísla napísané vedľa krúžkov predstavujú úroveň sebadôvery daného človeka.

Pre túto sociálnu sieť je najlepším riešením zadať anketu ľuďom 3 a 5. Súčet ich úrovní sebadôvery je $20 + 15 = 35$. Väčšiu hodnotu ako 35 už nevieme dosiahnuť.



Úloha

Napiš funkciu `findSample`. V jej parametroch dostaneš popis pridávania ľudí do sociálnej siete a ich úrovně sebadôvery. Tvoja funkcia má vypočítať a vrátiť najväčšiu možnú hodnotu súčtu úrovní sebadôvery množiny ľudí, z ktorých žiadni dvaja nie sú priatelia.

Zoznam parametrov funkcie `findSample`:

- `findSample(n, confidence, host, protocol)`
 - `n`: počet ľudí
 - `confidence`: pole s `n` prvkami; `confidence[i]` je úroveň sebadôvery človeka `i`.
 - `host`: pole s `n` prvkami; `host[i]` je hostiteľ v kroku `i`.
 - `protocol`: pole s `n` prvkami; `protocol[i]` je číslo udávajúce typ pozvánky použitý v kroku `i`: 0 je ČauKamoš, 1 je MámeRovnakýchKamošov a 2 je ČauKamošMámeRovnakýchKamošov.

- Keďže v kroku 0 nie je žiaden hostiteľ a neposiela sa žiadna pozvánka, `host[0]` a `protocol[0]` sú nedefinované. Tvoj program k nim nesmie pristupovať.
- Návrátovou hodnotou funkcie by mal byť najväčší možný súčet úrovni sebadôvery vybraných ľudí.

Podúlohy

Podúlohy sa líšia nielen počtom ľudí, ale aj rozsahom povolených hodnôt sebadôvery a tiež typmi pozvánok, ktoré sa v nich používajú. Všetko je prehľadne zhrnuté v nasledujúcej tabuľke.

podúloha	body	n	sebadôvera	typy pozvánok
1	11	$2 \leq n \leq 10$	$1 \leq \text{confidence} \leq 1,000,000$	všetky tri typy
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	len typ 1 (MámeRovnakýchKamošov)
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	len typ 2 (ČauKamošMáme...)
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	len typ 0 (ČauKamoš)
5	23	$2 \leq n \leq 1,000$	Všetky hodnoty confidence sú 1	typy 0 a 1 (ČauKamoš a MámeRovnakýchKamošov)
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{confidence} \leq 10,000$	všetky tri typy

Detaily implementácie

Odvzdávaš presne jeden súbor, nazvaný `friend.c`, `friend.cpp` alebo `friend.pas`. V tomto súbore by mala byť implementovaná vyššie popísaná funkcia `findSample`. Musí mať hlavičku uvedenú nižšie. Ak programuješ v C/C++, tvoj súbor musí vložiť (include) súbor `friend.h`.

C/C++

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

Pascal

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

Ukážkový grader

Ukážkový grader, ktorý máte k dispozícii, očakáva vstup v nasledovnom formáte:

- riadok 1: n
- riadok 2: `confidence[0], ..., confidence[n-1]`
- riadok 3: `host[1], protocol[1], host[2], protocol[2], ..., host[n-1], protocol[n-1]`. (Všimnite si, že hodnoty `host[0]` a `protocol[0]` neuvádzame.)

Ukážkový grader na výstup vypíše návratovú hodnotu tvojej funkcie `findSample`.