



الصديق

نريد بناء شبكة اجتماعية مؤلفة من n شخصاً مرقمين من $0, \dots, n-1$. بعض أزواج الأشخاص في هذه الشبكة سيكونون أصدقاء. إذا أصبح الشخص x صديقاً للشخص y , عندئذ يصبح y أيضاً صديقاً لـ x .

يتم إضافة الأشخاص إلى الشبكة عن طريق n مرحلة، والتي ترقم أيضاً من 0 إلى $n-1$. الشخص i تتم إضافته في المرحلة رقم i . بالمرحلة 0 , الشخص 0 يضاف بحيث أنه الشخص الوحيد الموجود ضمن الشبكة الاجتماعية هذه. في كل مرحلة من المراحل $n-1$ التالية، يتم إضافة شخص إلى الشبكة عن طريق شخص آخر مضيف، المضيف يمكن أن يكون أي شخص موجود ضمن الشبكة سابقاً. في المرحلة i ($0 < i < n$), يمكن للمضيف في تلك المرحلة أن يضيف شخصاً قداماً من i إلى الشبكة بإحدى الطرق الثلاثة التالية:

- "أنا صديقك" وتعني أن الشخص i يصبح صديق المضيف فقط.
- "أصدقائي هم أصدقاؤك" وتعني أن الشخص i يصبح صديقاً لكل شخص هو حالياً صديق للمضيف ولكن لاحظ أنه في هذه الطريقة لا يصبح الشخص i صديقاً للمضيف.
- "نحن أصدقاؤك" تجعل الشخص i صديقاً للمضيف وأيضاً صديقاً لكل شخص هو صديق للمضيف في اللحظة الحالية.

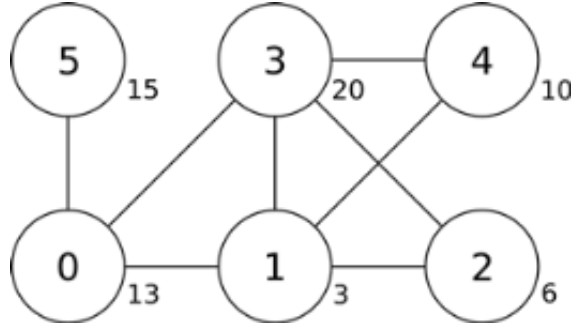
بعد بناء الشبكة نريد اختيار عينة من الأشخاص من أجل القيام باستطلاع رأي، أي اختيار مجموعة من الأشخاص من الشبكة، بحيث أن الأصدقاء عادة يكون لديهم اهتمامات مشتركة يجب على هذه العينة أن لا تحوي أي شخصين توجد بينهما صداقة. لكل شخص درجة ثقة في الاستبيانات يعبر عنها بعدد صحيح موجب، ونريد إيجاد العينة التي تحقق أعلى مجموع لدرجة الثقة هذه.

مثال

المرحلة	المضيف	الطريقة	علاقات الصداقة المضافة
1	0	أنا صديقك	(1, 0)
2	0	أصدقائي هم أصدقاؤك	(2, 1)
3	1	نحن أصدقاؤك	(3, 1), (3, 0), (3, 2)
4	2	أصدقائي هم أصدقاؤك	(4, 1), (4, 3)
5	0	أنا صديقك	(5, 0)

في البداية تحوي الشبكة الشخص رقم 0 ، في المرحلة رقم 1 يدعو المضيف (وهو الشخص رقم 0 حيث لا يوجد غيره في الشبكة) الشخص رقم 1 من خلال طريق "أنا صديقك"، وهكذا يصبحون أصدقاء. المضيف في المرحلة الثانية (وهو الشخص رقم 0 مرة أخرى) يدعو الشخص رقم 2 بطريقة "أصدقائي هم أصدقاؤك"، والذي يجعل الشخص رقم 1 (وهو الشخص الوحيد الصديق للمضيف) يجعله الصديق الوحيد للشخص رقم 2 ، يضيف المضيف في المرحلة الثالثة (وهو الشخص 1) الشخص رقم 3 من خلال عملية "نحن أصدقاؤك"، والتي تجعل الشخص رقم 3 صديقاً للشخص رقم 1 (وهو المضيف) وأيضاً للأشخاص 0 و 2 (أصدقاء المضيف).

المرحلتان 4 و 5 الموضحتان في الجدول أعلاه. يعرض الشكل التالي الشبكة النهائية حيث يعرض الرقم داخل الدائرة رقم الشخص والأرقام بجانب الدائرة تظهر درجة الثقة. في المثال عينة الأشخاص تحوي الشخصين 3 و 5 والذي يعطي درجة ثقة كلية تساوي $35=15+20$ ، والتي هي أعلى مجموع لدرجة الثقة الممكن الحصول عليها.



Task المهمة

بعد أعطائك وصف كل مرحلة ودرجة الثقة لكل شخص، المطلوب إيجاد العينة التي تحوي أعلى مجموع لدرجة الثقة. يجب عليك فقط كتابة الدالة `findSample`.

`findSample(n, confidence, host, protocol` ■

■ `n`: عدد الأشخاص .

■ `confidence`: مصفوفة طولها `n`; `confidence[i]` تعطي درجة الثقة للشخص `i`.

■ `host`: مصفوفة طولها `n`; `host[i]` تعطي المضيف للمرحلة `i`.

■ `protocol`: مصفوفة طولها `n`; `protocol[i]` تحدد رمز العملية التي تم القيام بها في المرحلة `i` ($0 < i < n$): 0 من أجل أنا صديقك, 1 for أصدقائي هم أصدقاؤك, and 2 for نحن أصدقاؤك.

■ بما أنه لا يوجد مضيف للمرحلة 0، لذلك فإن قيمة `host[0]` و `protocol[0]` غير محددة ولا يجب الوصول إليها من قبل برنامجك.

■ يجب أن ترجع الدالة أكبر قيمة ممكنة لمجموع درجة الثقة لعينة ما.

المهام الجزئية

بعض المسائل الجزئية تستخدم فقط مجموعة جزئية من العمليات، كما هو موضح في الجدول التالي

المسألة الجزئية	النقاط	n	درجة الثقة	الطريقة المستعملة
1	11	$2 \leq n \leq 10$	$1 \leq \text{confidence} \leq 1,000,000$	كل الطرق الثلاثة
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	"قط" أصدقائي هم أصدقاؤك
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	"فقط" نحن أصدقاؤك
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	"فقط" أنا صديقك
5	23	$2 \leq n \leq 1,000$	كل درجات الثقة تساوي 1	كل من "أصدقائي هم أصدقاؤك" و "أنا صديقك"
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{confidence} \leq 10,000$	كل الطرق الثلاثة

تفاصيل التنفيذ

يجب عليك إرسال ملف واحد `friend.cpp`.

يجب عليك تضمين المكتبة friend.h.

C/C++ program

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

Pascal programs

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

منظومة التصحيح

سنقوم منظومة التصحيح بقراءة الدخل وفقاً للتنسيق التالي:

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] ■

ستطبع منظومة التصحيح القيمة المرجعه من الداله findSample.