



## Friend

เราสร้างเครือข่ายสังคม (social network) จากผู้คน  $n$  คน แต่ละคนกำกับด้วยหมายเลข  $0, \dots, n - 1$  บางคู่ของบุคคลในเครือข่ายสังคม จะเป็นเพื่อนกัน ถ้าบุคคลหมายเลข  $x$  เป็นเพื่อนกับบุคคลหมายเลข  $y$  แล้ว บุคคลหมายเลข  $y$  ก็จะเป็นเพื่อนกับบุคคลหมายเลข  $x$  ด้วย

บุคคลต่าง ๆ จะถูกเพิ่มเข้าในเครือข่ายเป็นรอบ ๆ จำนวน  $n$  รอบ แต่ละรอบกำกับด้วยหมายเลขเป็น  $0$  ถึง  $n - 1$  เช่นเดียวกัน บุคคลที่  $i$  จะถูกเพิ่มในรอบที่  $i$  ในรอบที่  $0$  บุคคลหมายเลข  $0$  จะถูกเพิ่มเป็นบุคคลคนเดียวในเครือข่าย ในอีก  $n - 1$  รอบถัดไป บุคคลต่าง ๆ จะถูกเพิ่มเข้าในเครือข่ายโดยผ่านทางเจ้าภาพที่จะต้องเป็นบุคคลที่อยู่ในเครือข่ายแล้ว ในรอบที่  $i$  ( $0 < i < n$ ) เจ้าภาพของรอบนั้นจะสามารถเพิ่มบุคคลที่  $i$  เข้าไปในเครือข่าย ได้สามวิธีดังนี้:

- *IAmYourFriend* จะทำให้บุคคลที่  $i$  เป็นเพื่อนกับเจ้าภาพ
- *MyFriendsAreYourFriends* ทำให้บุคคลที่  $i$  เป็นเพื่อนกับบุคคลทุก ๆ คนที่เป็นเพื่อนของเจ้าภาพในเวลา นี้ สังเกตว่าวิธีนี้จะไม่ทำให้บุคคลที่  $i$  เป็นเพื่อนกับเจ้าภาพ
- *WeAreYourFriends* ทำให้บุคคลที่  $i$  เป็นเพื่อนกับเจ้าภาพและเป็นเพื่อนกับบุคคลทุก ๆ คนที่เป็นเพื่อนของเจ้าภาพในเวลา นี้

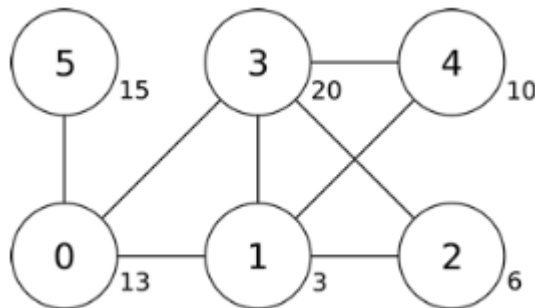
ภายหลังจากที่เราสร้างเครือข่ายแล้ว เราต้องการเลือกกลุ่มตัวอย่างเพื่อทำแบบสอบถาม นั่นคือ เลือกกลุ่มของบุคคลจากเครือข่าย เนื่องจากบุคคลที่เป็นเพื่อนกันมักมีความสนใจคล้าย ๆ กัน กลุ่มตัวอย่างนี้จะต้องไม่มีบุคคลสองคนใด ๆ ที่เป็นเพื่อนกัน นอกจากนี้ บุคคลแต่ละคนจะมีค่าความเชื่อมั่นของการทำแบบสอบถาม ที่จะมีค่าเป็นจำนวนเต็มบวก เราต้องการหากกลุ่มตัวอย่างที่มีผลรวมของค่าความเชื่อมั่นมากที่สุด

### ตัวอย่าง

รอบที่	เจ้าภาพ	วิธี	ความสัมพันธ์เพื่อนที่เพิ่มขึ้น
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)

เมื่อเริ่มต้นเครือข่ายมีแค่บุคคลที่ 0 เท่านั้น เจ้าภาพของรอบที่ 1 (บุคคลที่ 0) ได้เชิญบุคคลที่ 1 โดยใช้วิธี IAmYourFriend ทำให้บุคคลทั้งสองเป็นเพื่อนกัน เจ้าภาพของรอบที่ 2 (บุคคลที่ 0 เช่นเดิม) เชิญบุคคลที่ 2 ด้วยวิธี MyFriendsAreYourFriends ซึ่งทำให้บุคคลที่ 1 (ซึ่งเป็นเพื่อนคนเดียวของเจ้าภาพ) เป็นเพื่อนกับบุคคลที่ 2 ในรอบที่ 3 เจ้าภาพของรอบที่ 3 (บุคคลที่ 1) เพิ่มบุคคลที่ 3 ผ่านทางวิธี WeAreYourFriends ทำให้บุคคลที่ 3 เป็นเพื่อนกับบุคคลที่ 1 (เจ้าภาพ) บุคคลที่ 0 และบุคคลที่ 2 (เพื่อนของเจ้าภาพ) รอบที่ 4 และ 5 แสดงในตารางด้านบน เครือข่ายสุดท้ายแสดงในรูปถัดไป โดยที่หมายเลขในวงกลมแสดงหมายเลขของบุคคลต่าง ๆ ในเครือข่าย และจำนวนเต็มที่อยู่นอกวงกลมแสดงค่าความเชื่อมั่น กลุ่มตัวอย่างที่ประกอบด้วยบุคคลที่ 3 และบุคคลที่ 5 ที่มีผลรวมของค่าความเชื่อมั่นเท่ากับ  $20 + 15 = 35$  ซึ่งเป็นผลรวมค่าความเชื่อมั่นที่มากที่สุดเท่าที่เป็นไปได้



### งานของคุณ

เมื่อกำหนดรายละเอียดของแต่ละรอบและค่าความเชื่อมั่นของบุคคลแต่ละคน คำนวณหากลุ่มตัวอย่างที่มีผลรวมค่าความเชื่อมั่นมากที่สุด คุณต้องเขียนฟังก์ชัน `findSample`

- `findSample(n, confidence, host, protocol)`
  - `n`: จำนวนบุคคลในเครือข่าย
  - `confidence`: อาร์เรย์ความยาว `n` โดยที่ `confidence[i]` ระบุค่าความเชื่อมั่นของบุคคลที่ `i`
  - `host`: อาร์เรย์ความยาว `n` โดยที่ `host[i]` ระบุเจ้าภาพในรอบที่ `i`
  - `protocol`: อาร์เรย์ความยาว โดยที่ `protocol[i]` ระบุรหัสของวิธีที่ใช้ในรอบที่ `i`

( $0 < i < n$ ) ดังนี้ 0 แทน IAmYourFriend, 1 แทน MyFriendsAreYourFriends, และ 2 แทน WeAreYourFriends

- เนื่องจากในรอบที่ 0 ไม่มีเจ้าภาพ, `host[0]` และ `protocol[0]` จะไม่มีการระบุค่า (undefined) และต้องไม่ถูกใช้หรืออ่านโดยโปรแกรมของคุณ
- ฟังก์ชันต้องคืนผลรวมค่าความเชื่อมั่นที่มากที่สุด

## งานย่อย

บางงานย่อยใช้วิธีเพิ่มบุคคลแค่บางวิธีเท่านั้น ดังแสดงในตารางด้านล่าง

งานย่อย	คะแนน	$n$	ค่าความเชื่อมั่น	วิธีการเพิ่มบุคคล
1	11	$2 \leq n \leq 10$	$1 \leq \text{confidence} \leq 1,000,000$	ทุกวิธี
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	MyFriendsAreYourFriends เท่านั้น
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	WeAreYourFriends เท่านั้น
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	IAmYourFriend เท่านั้น
5	23	$2 \leq n \leq 1,000$	ค่าความเชื่อมั่นทุกค่ามีค่าเป็น 1	MyFriendsAreYourFriends และ IAmYourFriend
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{confidence} \leq 10,000$	ทุกวิธี

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องส่งแฟ้มหนึ่งแฟ้ม โดยมีชื่อเป็น `friend.c`, `friend.cpp` หรือ `friend.pas` โดยที่แฟ้มนี้จะต้องเขียนโปรแกรมย่อยดังที่กล่าวไว้ข้างต้น โดยใช้รูปแบบดังต่อไปนี้ และคุณจำเป็นที่จะต้อง include header `friend.h` ในแฟ้มดังกล่าวด้วยถ้าคุณใช้ภาษา C/C++

โปรแกรมภาษา C/C++

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

## โปรแกรมภาษา Pascal

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

### เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะอ่านข้อมูลนำเข้าดังรูปแบบต่อไปนี้

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

เกรตเตอร์ตัวอย่างจะพิมพ์ค่าที่คืนจากฟังก์ชัน **findSample**