**★★★☆☆**

**題號：Q12442: Forwarding Emails**

**整理者：陳紫淇**

**學號：ADT105142**

**使用語言:C++**

**解題日期：2019年1月7日**



**題目:**

“... so forward this to ten other people, to prove that you believe the emperor has new clothes.” Aren’t those sorts of emails annoying? Martians get those sorts of emails too, but they have an innovative way of dealing with them. Instead of just forwarding them willy-nilly, or not at all, they each pick one other person they know to email those things to every time - exactly one, no less, no more (and never themselves). Now, the Martian clan chieftain wants to get an email to start going around, but he stubbornly only wants to send one email. Being the chieftain, he managed to ﬁnd out who forwards emails to whom, and he wants to know: which Martian should he send it to maximize the number of Martians that see it?

**Input**

The ﬁrst line of input will contain T (≤ 20) denoting the number of cases. Each case starts with a line containing an integer N (2 ≤ N ≤ 50000) denoting the number of Martians in the community. Each of the next N lines contains two integers: u v (1 ≤ u,v ≤ N, u ̸= v)

**Output**

For each case, print the case number and an integer m, where m is the Martian that the chieftain should send the initial email to. If there is more than one correct answer, output the smallest number.

**Sample Input**

3

3

1 2

2 3

3 1

4

1 2

2 1

4 3

3 2

5

1 2

2 1

5 3

3 4

4 5

**Sample Output**

Case 1: 1

Case 2: 4

Case 3: 3

**問題描述：**

「...務必轉寄給十個人，以證明你相信國王有新衣。」

 這種 email 很討人厭，不是嗎？

火星人也有這種郵件，但他們有個新奇的方式來處理它。他們既不亂寄，也不會不寄，而是**只寄給一個朋友**，不多也不少，(而且不會寄給自己)。現在火星部落酋長要發一封 email 出去，他很固執只肯發給一個人。身為酋長，他設法查出了誰會轉信給誰，現在他想知道：他的信要寄給誰才能讓最多的火星人看到？

**Input**

輸入的第一行有一個 **T (≤ 20)** 表示測資筆數。

每筆測資的第一行有一個整數 **N (2 ≤ N ≤ 50000)** 表示社群中火星人的數量。以下 **N** 行每行有兩個整數：**u v (1 ≤ u, v ≤ N, u ≠ v)** 代表火星人 **u** 會把 email 轉給火星人 **v**。

**Output**

對於每筆測資，印出測資編號及一個整數 **m**，代表酋長應該把初始郵件寄達的那個火星人。如果正確答案不止一個，輸出最小的數字。

**Sample Input**

3

3

1 2

2 3

3 1

4

1 2

2 1

4 3

3 2

5

1 2

2 1

5 3

3 4

4 5

**Sample Output**

Case 1: 1

Case 2: 4

Case 3: 3

**解法:**

1. 使用遞迴函式追蹤信件的去向
2. 將信件接觸過的所有人記錄下來，以後不對這些人追蹤信件，因為一號之後的人不可能寄的比一號還多，可以跳過他們
3. 排除步驟2記錄過的人，重複步驟1 2 3

**解法範例：**

1. 建立遞迴函式，不斷的找下一個會寄給誰，

用陣列hasMail紀錄**這次**已收過信的人，

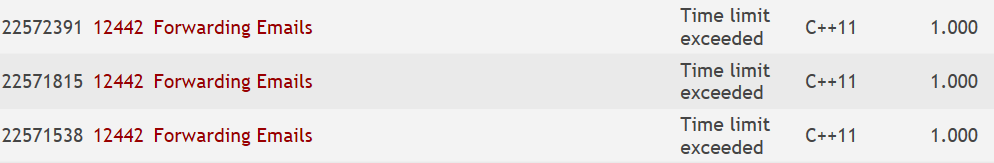
陣列done紀錄**所有**曾經接觸信件的人，

直到出現這次重複收信的人為止，回傳人數並比計較是否比目前多人數多，若是則更新寄出信的第一個人。

1. 找到下一個沒接觸過信件的人(done沒有紀錄)，重複1

**討論:**

1. 如果不排除接觸過信件的人，會超過時間



|  |
| --- |
| #include <iostream>  #include <cstdio>  #include <vector>  using namespace std;  vector<bool> hasMail;  vector<int> addr;  vector<bool> done;  int trace(int m)  {  hasMail[m] = true;  if(hasMail[addr[m]]==true) //收件人有收過信  return 0;  else  {  done[m]=true;  return 1+trace(addr[m]);  }  }  int main()  {  int T,N;  int u,v;  int CASE = 1;  cin>>T; //題目數量  while(T--)  {  cin>>N; //火星人的數量  addr.assign(N, -1); //誰寄給誰  for(int i=0;i<N;i++)  {  cin>>u>>v;  addr[u-1] = v-1;  }  done.assign(N, false);  int MAX = 1;  int firstMan = 1;  for(int j=0;j<N;j++)  {  if(done[j]) //做過了  continue;  hasMail.assign(N, false);  int NEW = trace(j);  if( NEW > MAX )  {  firstMan = j;  MAX = NEW;  }  }  printf("Case %d: %d\n",CASE,firstMan+1);  CASE++;  }  } |