

სოციალური ინჟინერია

Problem Name	Social Engineering
Input file	Interactive task
Output file	Interactive task
Time limit	5 seconds
Memory limit	256 megabytes

სოციალური ქსელი წარმოადგენს არაორიენტირებულ ბმულ გრაფს n წვეროსა და m წიბოსაგან, სადაც ყოველი წვერო წარმოადგენს ადამიანს, ხოლო ორი ადამიანი ითვლება მეგობრად, თუ მათ შორის არსებობს წიბო. მარია არის სოციალური ქსელის წევრი. მას უყვარს ჩელენჯების (გამონწვევა, გამოცდა) მოწყობა თავისი მეგობრებისთვის. ეს ნიშნავს, რომ თავდაპირველად ის ასრულებს რაიმე მარტივ ამოცანას და შემდეგ იწვევს ერთ-ერთ თავის მეგობარს იგივეს გასაკეთებლად. ამის შემდეგ ეს ჩელენჯი მოგზაურობს ქსელში და შესაძლოა ერთი და იგივე ადამიანმა იგი რამდენჯერმე მიიღოს, ოღონდ მეგობრების ყოველ დაულაგებელ წევრს ჩელენჯში მონაწილეობა მხოლოდ ერთხელ შეუძლია (როგორც კი A ადამიანი გამოიწვევს B ადამიანს, არც A -ს და არც B -ს აღარ შეუძლიათ ერთმანეთი გამოიწვიონ). სხვა სიტყვებით, ჩელენჯი წარმოადგენს გრაფის შემოვლას, რომელიც ვერ გამოიყენებს ვერცერთ წიბოს 1-ზე მეტად. ადამიანი აგებს ჩელენჯს, თუ ის ვერცერთ თავის მეგობარს ვეღარ გამოიწვევს. ჩელენჯს ყოველთვის იწყებს მარია და მას ჯერ არასდროს არ წაუგია. ამჟერად, დარჩენილმა $n - 1$ ადამიანმა გადაწყვიტა ითანამშრომლოს, რათა მარიამ წააგოს მორიგი ჩელენჯი. თქვენი მიზანია, კოორდინირება გაუკეთოთ მათ მოქმედებებს.

იმპლემენტაცია

თქვენ უნდა მოახდინოთ შემდეგი ფუნქციის იმპლემენტაცია:

```
void SocialEngineering(int n, int m, vector<pair<int,int>> edges);
```

რომელიც ითამაშებს n წვეროსა და m წიბოსაგან შედგენილ გრაფზე. ეს ფუნქცია გრადერის მიერ გამოიძახება ერთხელ. სია `edges` შეიცავს მთელი დადებითი რიცხვების ბუსტად m წყვილს (u, v) , რაც ნიშნავს, რომ არსებობს წიბო u და v წვეროებს შორის. წვეროები გადანომრილია 1-დან n -მდე. მარია ყოველთვის წარმოადგენს 1 წვეროს. თქვენმა ფუნქციამ უნდა გამოიძახოს შემდეგი მეთოდები:

```
int GetMove();
```

ეს მეთოდი უნდა გამოიძახოთ ყოველთვის, როცა არის მარიას სვლა (მაგალითად, თამაშის

დანყებისას) თუ თქვენ გამოიძახებთ ამ ფუნქციას მაშინ, როცა მარიას სვლა არ არის, თქვენ მიიღებთ ვერდიქტს `Wrong Answer`. ფუნქციამ უნდა დააბრუნოს შემდეგი მნიშვნელობებიდან ერთ-ერთი:

- მთელი დადებითი რიცხვი v , სადაც $2 \leq v \leq n$. ეს ნიშნავს, რომ მარია ჩელენჯს უგზავნის v ინდექსის მქონე ადამიანს. ეს სვლა ყოველთვის ლეგალურია.
- 0, თუ მარიამ წააგო თამაში. მარია ყოველთვის წყვეტს თამაშს, თუკი მას არ აქვს ლეგალური სვლა. როცა ეს მოხდება თქვენმა პროგრამამ უნდა დააბრუნოს ფუნქცია `SocialEngineering()`, და თქვენ მიიღებთ ვერდიქტს: `Accepted`.

```
void MakeMove(int v);
```

ეს მეთოდი გამოიძახებულ უნდა იქნას ყოველთვის, როცა არ არის მარიას სვლა. ეს ნიშნავს, რომ ჩელენჯის შემსრულებელი გამოწვევას უგზავნის v ნომრის მქონე ადამიანს. თუ ეს სვლა არ არის ლეგალური, ან ამ მომენტში სვლა მარიას ეკუთვნოდა, თქვენ მიიღებთ ვერდიქტს `Wrong Answer`. თუ მარიას აქვს მომგებიანი სტრატეგია თამაშის დასაწყისშივე, თქვენმა პროგრამამ ნება უნდა დართოს `SocialEngineering()`-ს დაბრუნდეს `GetMove()`-ს პირველ გამოძახებამდე. ამ შემთხვევაში თქვენ მიიღებთ ვერდიქტს `Accepted`.

შეზღუდვები

- $2 \leq n \leq 2 \cdot 10^5$.
- $1 \leq m \leq 4 \cdot 10^5$.
- გრაფი ბმულია. ყოველი დაულაგებელი წყვილი ნიბოთა სიაში გვხვდება არაუმეტეს 1-სა. ყოველი ნიბო აერთებს ორ განსვავებულ ნიბოს.

ქვეამოცანები

მარია ყოველთვის ითამაშებს ოპტიმალურად, ანუ ყოველთვის გააკეთებს მომგებიან სვლებს, თუ მას აქვს მომგებიანი სტრატეგია. თუ მას მომგებიანი სტრატეგია არ აქვს, მარია ეცდება სხვადასხვა საშუალებით დააშვებინოს შეცდომა თქვენს პროგრამას. ის თამაშს შეწყვეტს (ანუ წააგებს) მხოლოდ მაშინ, როცა მას არ ექნება ლეგალური სვლები გარდა მესამე ქვეამოცანისა.

1. (15 ქულა) $n, m \leq 10$.
2. (15 ქულა) მარიას გარდა, ყველას ჰყავს არაუმეტეს 2 მეგობარი.
3. (20 ქულა) მარია დაუყოვნებლივ აგებს, თუკი მას არ აქვს მომგებიანი სტრატეგია.
4. (25 ქულა) $n, m \leq 100$.

5. (25 ქულა) დამატებითი შეზღუდვების გარეშე.

ინტერაქციის მაგალითი

Contestant action	Grader action	Explanation
-	SocialEngineering(5, 6, {{1,4}, {1,5}, {2,4}, {2,5}, {2,3}, {3,5}})	SocialEngineering() გამოძახებულია გრაფისთვის 5 წვერო და 6 ნიბო.
GetMove()	Returns 4	მარიამ ჩელენჯი გაუგზავნა ადამიანს 4.
MakeMove(2)	-	ადამიანმა 4 გამოიწვია ადამიანი 2.
MakeMove(5)	-	ადამიანმა 2 გამოიწვია ადამიანი 5.
MakeMove(1)	-	ადამიანმა 2 გამოიწვია მარია.
GetMove()	Returns 0	მარიას არ აქვს ლეგალური სვლები, ამიტომ მან შეწყვიტა თამაში და წააგო.
Returns	-	მონაწილემ მოიგო თამაში და მან SocialEngineering()-ს დაბრუნების საშუალება უნდა მისცეს.

Contestant action	Grader action	Explanation
-	SocialEngineering(2, 1, {{1,2}})	SocialEngineering გამოძახებულია გრაფისთვის 2 წვეროსა და 1 ნიბოსთვის.
Returns	-	მარიას აქვს მომგებიანი სტრატეგია და მონაწილე სასწრაფოდ უნდა დაბრუნდეს თამაშის შესაწყვეტად.

სანიმუშო გრაფერი

გრაფერის წარმოდგენილი ნიმუში grader.cpp, რომელიც მოცემულია ამოცანის ბმულში SocialEngineering.zip, კითხულობს შემოსატან მონაცემებს შემდეგი ფორმატით:

- პირველი სტრიქონი შეიცავს წვეროების რაოდენობას n , და ნიბოების რაოდენობას m , გრაფში.
- მომდევნო m სტრიქონი შეიცავს ნიბოების აღწერას წვეროთა ნყვილით.

გრაფერი კითხულობს შესატან მონაცემებს და იძახებს მონაწილის მიერ დაწერილ ფუნქციას SocialEngineering(). მიაქციეთ ყურადღება, რომ სანიმუშო გრაფერში არ არის იმპლემენტირებული მარიას მომგებიანი სტრატეგია და ის განკუთვნილია მხოლოდ ინტერაქციის

გამართვისათვის. იმისათვის, რომ დააკომპილიროთ თქვენი ამოხსნა სანიმუშო გრადერთან ერთად, საჭიროა გაუშვათ შემდეგი ბრძანება ბრძანებათა ტერმინალიდან:

```
g++ -std=gnu++11 -O2 -o solution grader.cpp solution.cpp
```

სადაც `solution.cpp` არის თქვენი ამოხსნა, რომელიც უნდა გააგზავნოთ CMS-ზე. თუ გსურთ, გატესტოთ პროგრამა ბმულში მოცემულ მაგალითზე, ბრძანებათა ტერმინალში მიუთითეთ:

```
./solution < input.txt
```