

სფინქსის გამოცანა

დიდ სფინქსს თქვენთვის გამოცანა აქვს. მოცემული გაქვთ N წვეროიანი გრაფი. წვეროები გადანომრილია 0-დან $(N - 1)$ -მდე. გრაფში არის M ცალი წიბო, დანომრილი 0-დან $(M - 1)$ -მდე. თითოეული წიბო არის ორმხრივი და აკავშირებს განსხვავებული წვეროების წყვილს. კონკრეტულად, თითოეული j -ისათვის 0-დან $(M - 1)$ -მდე (ჩათვლით), წიბო j აკავშირებს $X[j]$ და $Y[j]$ წვეროებს. არსებობს მაქსიმუმ ერთი წიბო, რომელიც აკავშირებს წვეროების ნებისმიერ წყვილს. ორ წვეროს ეწოდება **მეზობელი**, თუ ისინი დაკავშირებულია წიბოთი.

წვეროთა მიმდევრობას v_0, v_1, \dots, v_k ($k \geq 0$) ეწოდება **ბილიკი**, თუ ყოველი ორი მომდევნო წვერო v_l და v_{l+1} (თითოეული l -სათვის, რომლისთვისაც $0 \leq l < k$) არიან მეზობლები. ჩვენ ვამბობთ, რომ ბილიკი v_0, v_1, \dots, v_k **აკავშირებს** წვეროებს v_0 და v_k . თქვენთვის მოცემულ გრაფში წვეროების თითოეული წყვილი დაკავშირებულია გარკვეული ბილიკით.

არსებობს $N + 1$ განსხვავებული ფერი, დანომრილი 0-დან N -მდე. ფერი N განსაკუთრებულია და მას **სფინქსის ფერი** ეწოდება. თითოეულ წვეროს აქვს საკუთარი ფერი. კონკრეტულად, i ($0 \leq i < N$) წვეროს აქვს ფერი $C[i]$. სხვადასხვა წვეროებს შეიძლება ჰქონდეთ ერთი და იგივე ფერი. ასევე შეიძლება ზოგიერთი ფერი არ იყოს მინიჭებული არცერთ წვეროზე. არცერთ წვეროს არ აქვს სფინქსის ფერი, ანუ $0 \leq C[i] < N$ ($0 \leq i < N$).

ბილიკს v_0, v_1, \dots, v_k ($k \geq 0$) ეწოდება **მონოქრომატული**, თუ მის ყველა წვეროს აქვს ერთი და იგივე ფერი, ანუ $C[v_l] = C[v_{l+1}]$ (თითოეული ისეთი l -ისთვის, რომ $0 \leq l < k$). გარდა ამისა, ჩვენ ვამბობთ, რომ წვეროები p და q ($0 \leq p < N$, $0 \leq q < N$) არიან ერთი და იგივე **მონოქრომატულ კომპონენტში** მაშინ და მხოლოდ მაშინ, როცა ისინი დაკავშირებული არიან მონოქრომატული ბილიკით.

თქვენ იცით წვეროები და წიბოები, მაგრამ არ იცით რა ფერი აქვს თითოეულ წვეროს. გსურთ გაიგოთ წვეროების ფერები, **გადაღების ექსპერიმენტების ჩატარებით**.

გადაღების ექსპერიმენტში თქვენ შეგიძლიათ სურვილისამებრ გადააფერადოთ ნებისმიერი რაოდენობის წვერო. კერძოდ, გადაღების ექსპერიმენტის ჩასატარებლად თქვენ ჯერ ირჩევთ N ზომის მასივს E , სადაც თითოეული i -სთვის ($0 \leq i < N$), $E[i]$ არის -1 -დან N -ის **ჩათვლით**. შემდეგ, თითოეული i წვეროს ფერი ხდება $S[i]$, სადაც $S[i]$ არის:

- $C[i]$, ანუ i -ს ორიგინალური ფერი, თუ $E[i] = -1$, ან
- $E[i]$ წინააღმდეგ შემთხვევაში.

შევნიშნოთ, რომ ეს ნიშნავს, რომ გადაღებისას შეგიძლიათ სფინქსის ფერის გამოყენება.

ბოლოს, დიდი სფინქსი გეუბნებათ მონოქრომატული კომპონენტების რაოდენობას გრაფში თითოეული i წვეროსთვის $S[i]$ ($0 \leq i < N$) ფერის მინიჭების შემდეგ. ახალი გადაღება გამოიყენება მხოლოდ ამ კონკრეტული გადაღების ექსპერიმენტისთვის. ანუ, **თითოეული წვეროს ფერი უბრუნდება თავდაპირველს ექსპერიმენტის დასრულების შემდეგ.**

თქვენი ამოცანაა დაადგინოთ წვეროების ფერები გრაფში მაქსიმუმ 2 750 გადაღების ექსპერიმენტების ჩატარებით. თქვენ ასევე შეგიძლიათ მიიღოთ ნაწილობრივი ქულები, თუ მეზობელი წვეროების ყველა წყვილისთვის სწორად განსაზღვრავთ, აქვთ თუ არა მათ ერთი და იგივე ფერი.

იმპლემენტაციის დეტალები

თქვენ უნდა მოახდინოთ შემდეგი პროცედურის იმპლემენტაცია.

```
std::vector<int> find_colours(int N,  
std::vector<int> X, std::vector<int> Y)
```

- N : წვეროების რაოდენობა გრაფში;
- $X, Y : M$ სიგრძის მასივები, რომლებიც აღწერენ წიბოებს;
- ამ პროცედურამ უნდა დააბრუნოს N სიგრძის მასივი G , რომელიც წარმოადგენს გრაფის წვეროების ფერებს;
- ეს პროცედურა გამოიძახებული იქნება ზუსტად ერთხელ თითოეული ტესტისათვის.

ზემოთ მოცემულ პროცედურას შეუძლია გამოიძახოს შემდეგი პროცედურა გადაღების ექსპერიმენტების ჩასატარებლად:

```
int perform_experiment(std::vector<int> E)
```

- $E : N$ სიგრძის მასივი, რომელიც აღწერს, თუ როგორ უნდა გადაიღებოს წვეროები;
- ეს პროცედურა აბრუნებს მონოქრომატული კომპონენტების რაოდენობას E -ს მიხედვით წვეროების გადაღების შემდეგ;
- ამ პროცედურის გამოიძახება შესაძლებელია მაქსიმუმ 2 750-ჯერ.

გრაფერი არ არის ადაპტიური. ანუ, წვეროების ფერები ფიქსირებულია find_colours-ის გამოიძახებამდე .

შეზღუდვები

- $2 \leq N \leq 250$
- $N - 1 \leq M \leq \frac{N \cdot (N - 1)}{2}$
- $0 \leq X[j] < Y[j] < N$ თითოეული ისეთი j -სათვის, რომ $0 \leq j < M$
- $X[j] \neq X[k]$ ან $Y[j] \neq Y[k]$ სრულდება თითოეული ისეთი j და k -სათვის, რომ $0 \leq j < k < M$

- წვეროების თითოეული წყვილი დაკავშირებულია გარკვეული ბილიკით
- $0 \leq C[i] < N$ ყოველი ისეთი i -სთვის, რომ $0 \leq i < N$.

ქვეამოცანა

ქვეამოცანა	ქულა	დამატებითი შეზღუდვები
1	3	$N = 2$
2	7	$N \leq 50$
3	33	გრაფი არის ბილიკი: $M = N - 1$ და წვეროები j და $j + 1$ არიან მეზობლები ($0 \leq j < M$)
4	21	გრაფი არის სრული: $M = \frac{N \cdot (N-1)}{2}$ და წვეროთა ნებისმიერი წყვილი არის მეზობელი
5	36	დამატებითი შეზღუდვების გარეშე

თითოეულ ქვეამოცანაში შეგიძლიათ მიიღოთ ნაწილობრივი ქულა, თუ თქვენი პროგრამა სწორად განსაზღვრავს მეზობელი წვეროების ყოველი წყვილისთვის აქვთ თუ არა მათ ერთი და იგივე ფერი.

უფრო ზუსტად, თქვენ იღებთ ქვეამოცანის მთელ ქულას, თუ მის ყველა ტესტში find_colours-ის მიერ დაბრუნებული G მასივი ზუსტად იგივეა, რაც C მასივი (ანუ $G[i] = C[i]$ ყველა ისეთი i -სთვის, რომ $0 \leq i < N$). წინააღმდეგ შემთხვევაში თქვენ მიიღებთ ქულების 50%-ს ქვეამოცანისთვის, თუ სრულდება შემდეგი პირობები მის ყველა ტესტში:

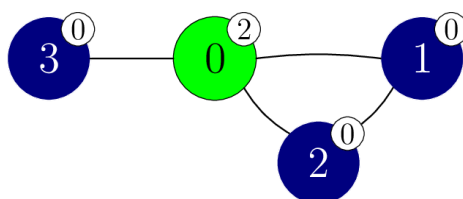
- $0 \leq G[i] < N$ თითოეული ისეთი i -სთვის, რომ $0 \leq i < N$;
- თითოეული ისეთი j -სთვის, რომ $0 \leq j < M$:
 - $G[X[j]] = G[Y[j]]$ მაშინ და მხოლოდ მაშინ, როცა $C[X[j]] = C[Y[j]]$.

მაგალითი

განვიხილოთ შემდეგი გამოცახება.

```
find_colours(4, [0, 1, 0, 0], [1, 2, 2, 3])
```

ამ მაგალითისთვის, დავუშვათ, რომ წვეროების (დამალული) ფერები მოიცემა $C = [2, 0, 0, 0]$ მასივით. ეს სცენარი ნაჩვენებია ქვემოთ მოცემულ სურათში. ფერები დამატებით წარმოდგენილია ციფრებით თეთრ წრეებში, რომლებიც მიმაგრებულია თითოეულ წვეროზე.



პროცედურას შეუძლია გამოიძახოს `perform_experiment` შემდეგნაირად.

```
perform_experiment([-1, -1, -1, -1])
```

ამ ექსპერიმენტში არცერთი წვერო არ არის შეღებილი. შესაბამისად, ყველა წვერო ინარჩუნებს თავდაპირველ ფერს.

განვიხილოთ წვერო 1 და წვერო 2. ორივეს აქვს ფერი 0 და ბილიკი 1, 2 არის მონოქრომატული ბილიკი. შედეგად, 1 და 2 წვეროები ერთსა და იმავე მონოქრომატულ კომპონენტშია.

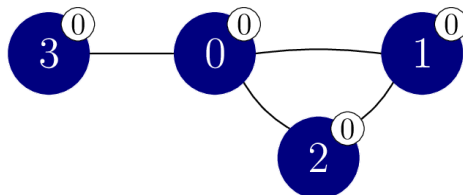
განვიხილოთ წვერო 1 და წვერო 3. მიუხედავად იმისა, რომ ორივეს აქვს ფერი 0, ისინი სხვადასხვა მონოქრომატულ კომპონენტებში არიან, რადგან არ არსებობს მათი დამაკავშირებელი მონოქრომატული ბილიკი.

საერთო ჯამში, არის 3 მონოქრომატული კომპონენტი $\{0\}$, $\{1, 2\}$ და $\{3\}$ წვეროებით. ამრიგად, ეს პროცედურა აბრუნებს 3-ს.

ახლა, პროცედურამ შეიძლება გამოიძახოს `perform_experiment` შემდეგნაირად.

```
perform_experiment([0, -1, -1, -1])
```

ამ გამოძახებისას მხოლოდ 0 წვერო გადაფერადდება 0 ფერად, რის შედეგადაც მიიღება შემდეგ სურათზე ნაჩვენები შედეგა.

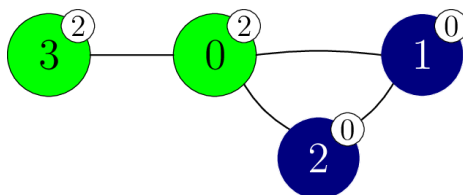


ეს გამოძახება აბრუნებს 1-ს, რადგან ყველა წვერო ეკუთვნის ერთსა და იმავე მონოქრომატულ კომპონენტს. აქედან შეგვიძლია დავასკვნათ, რომ 1, 2 და 3 წვეროებს აქვთ ფერი 0.

შემდეგ, პროცედურამ შეიძლება გამოიძახოს `perform_experiment` შემდეგნაირად.

```
perform_experiment([-1, -1, -1, 2])
```

ამ გამოძახებისას, წვერო 3 გადაიღებება ფერად 2, რის შედეგადაც მიიღება შემდეგ სურათზე ნაჩვენები შედეგა.



ეს გამოძახება დააბრუნებს 2-ს, რადგან არსებობს 2 მონოქრომატული კომპონენტი, წვეროებით $\{0, 3\}$ და $\{1, 2\}$. შეგვიძლია დავასკვნათ, რომ წვეროს 0 აქვს ფერი 2.

პროცედურა `find_colours` აბრუნებს მასივს $[2, 0, 0, 0]$. ვინაიდან $C = [2, 0, 0, 0]$, მიღებული იქნება სრული ქულა.

გაითვალისწინეთ, რომ ასევე არსებობს არაერთი შესაძლო დაბრუნებული მნიშვნელობა, რომელიც მიიღებდა ქულების 50%-ს, მაგალითად $[1, 2, 2, 2]$ ან $[1, 2, 2, 3]$.

გრადერის ნიმუში

შეტანის ფორმატი:

```
NM
C[0] C[1] ... C[N-1]
X[0] Y[0]
X[1] Y[1]
...
X[M-1] Y[M-1]
```

გამოტანის ფორმატი:

```
LQ
G[0] G[1] ... G[L-1]
```

აქ L არის G მასივის სიგრძე, რომელიც `find_colours`-მა დააბრუნა და Q არის `perform_experiment`-ის გამოძახებების რაოდენობა.