

معمای ابوالهول

ابوالهول بزرگ معمایی برای شما دارد. به شما یک گراف با N رأس داده می‌شود. رأس‌ها از 0 تا $N - 1$ شماره‌گذاری شده‌اند. در این گراف، M یال وجود دارد که از 0 تا $M - 1$ شماره‌گذاری شده‌اند. هر یال یک زوج رأس متمایز را متصل می‌کند و دوطرفه است. به طور خاص، برای هر j از 0 تا $M - 1$ (شامل هر دو)، یال j رأس‌های $X[j]$ و $Y[j]$ را متصل می‌کند. هر دو رأس توسط حداکثر یک یال به هم متصل می‌شوند. دو رأس مجاور نامیده می‌شوند اگر با یک یال به هم متصل شده باشند.

دنباله‌ای از رأس‌ها v_0, v_1, \dots, v_k (برای $k \geq 0$) **مسیر** نامیده می‌شود اگر هر دو رأس متوالی v_l و v_{l+1} (برای هر l که $0 \leq l < k$) مجاور باشند. می‌گوییم که مسیر v_0, v_1, \dots, v_k رأس‌های v_0 و v_k را **متصل می‌کند**. در گراف داده شده به شما، هر جفت از رأس‌ها توسط حداقل یک مسیر به هم متصل می‌شوند.

$N + 1$ رنگ وجود دارد که از 0 تا N شماره‌گذاری شده‌اند. رنگ N خاص است و **رنگ ابوالهول** نامیده می‌شود. به هر رأس یک رنگ اختصاص داده شده است. به طور خاص، رأس i ($0 \leq i < N$) رنگ $C[i]$ را دارد. چندین رأس ممکن است یک رنگ مشترک داشته باشند، و ممکن است رنگ‌هایی وجود داشته باشد که به هیچ رأسی اختصاص داده نشده باشند. هیچ رأسی رنگ ابوالهول را ندارد، یعنی $0 \leq C[i] < N$ ($0 \leq i < N$).

مسیر v_0, v_1, \dots, v_k (برای $k \geq 0$) **تک‌رنگ** نامیده می‌شود اگر تمام رأس‌های آن یک رنگ داشته باشند، یعنی $C[v_l] = C[v_{l+1}]$ (برای هر l که $0 \leq l < k$). همچنین، می‌گوییم که رأس‌های p و q ($0 \leq p < N, 0 \leq q < N$) در یک **مولفه تک‌رنگ** قرار دارند اگر و تنها اگر توسط یک مسیر تک‌رنگ به هم متصل شوند.

شما رأس‌ها و یال‌ها را می‌دانید، اما نمی‌دانید هر رأس چه رنگی دارد. شما می‌خواهید رنگ رأس‌ها را با انجام **آزمایش‌های رنگ‌آمیزی مجدد** بیابید.

در یک آزمایش رنگ‌آمیزی مجدد، می‌توانید هر تعداد رأسی را که می‌خواهید دوباره رنگ‌آمیزی کنید. به طور خاص، برای انجام یک آزمایش رنگ‌آمیزی مجدد ابتدا یک آرایه E به اندازه N انتخاب می‌کنید، که برای هر i ($0 \leq i < N$)، $E[i]$ بین -1 و N **شامل هر دو** است. سپس، رنگ هر رأس i به $S[i]$ تبدیل می‌شود، که مقدار $S[i]$ به شرح زیر است:

- $C[i]$ ، یعنی رنگ اصلی i ، اگر $E[i] = -1$ باشد، یا
- $E[i]$ ، در غیر این صورت.

توجه کنید که این بدان معناست که می‌توانید از رنگ ابوالهول در رنگ‌آمیزی مجدد خود استفاده کنید.

در نهایت، ابوالهول بزرگ تعداد مولفه‌های تک‌رنگ در گراف را اعلام می‌کند، پس از تغییر رنگ هر رأس i به $S[i]$ ($0 \leq i < N$). رنگ‌آمیزی جدید فقط برای این آزمایش رنگ‌آمیزی مجدد اعمال می‌شود، بنابراین رنگ تمام رأس‌ها پس از پایان آزمایش به حالت اصلی باز می‌گردد.

وظیفه شما این است که رنگ رأس‌های گراف را با انجام حداکثر 2750 آزمایش رنگ‌آمیزی مجدد شناسایی کنید. همچنین ممکن است نمره جزئی دریافت کنید اگر برای هر جفت رأس مجاور به درستی تعیین کنید که آیا رنگ یکسانی دارند یا خیر.

جزئیات پیاده‌سازی

شما باید تابع زیر را پیاده‌سازی کنید.

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

- N : تعداد رأس‌های موجود در گراف.
- X, Y : آرایه‌هایی با طول M که یال‌ها را توصیف می‌کنند.
- این تابع باید آرایه G با طول N را بازگرداند، که نمایانگر رنگ‌های رأس‌های گراف است.
- این تابع دقیقاً یک بار برای هر تست کیس فراخوانی می‌شود.

تابع بالا می‌تواند برای انجام آزمایش‌های رنگ‌آمیزی مجدد، تابع زیر فراخوانی کند:

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

- E : یک آرایه با طول N که تعیین می‌کند چگونه رأس‌ها باید دوباره رنگ‌آمیزی شوند.
- این تابع تعداد مولفه‌های تک‌رنگ را پس از رنگ‌آمیزی مجدد رأس‌ها مطابق با E بازمی‌گرداند.
- این تابع حداکثر می‌تواند 2,750 بار فراخوانی شود.

ارزیاب **انطباقی نیست**، به این معنا که رنگ‌های رأس‌ها قبل از فراخوانی به `find_colours` ثابت هستند.

محدودیت‌ها

- $2 \leq N \leq 250$
- $N - 1 \leq M \leq \frac{N \cdot (N-1)}{2}$
- $0 \leq j < M$ برای هر j به طوری که $0 \leq X[j] < Y[j] < N$
- $X[j] \neq X[k]$ یا $Y[j] \neq Y[k]$ برای هر j و k به طوری که $0 \leq j < k < M$.
- هر جفت رأس توسط یک مسیر به هم متصل هستند.
- $0 \leq i < N$ برای هر i به طوری که $0 \leq C[i] < N$.

زیرمسئله‌ها

زیرمسئله	امتیاز	محدودیت‌های اضافی
1	3	$N = 2$
2	7	$N \leq 50$
3	33	گراف یک مسیر است: $M = N - 1$ و رأس‌های z و $z + 1$ مجاور هستند
4	21	گراف کامل است: $M = \frac{N \cdot (N-1)}{2}$ و هر دو رأس مجاور هستند
5	36	بدون محدودیت‌های اضافی

در هر زیرمسئله، اگر برنامه شما به درستی تعیین کند برای هر جفت رأس مجاور آیا آنها رنگ یکسانی دارند یا خیر، می‌توانید امتیاز جزئی دریافت کنید.

به طور دقیق‌تر، شما امتیاز کامل یک زیرمسئله را دریافت می‌کنید اگر در همه تست کیس‌های آن، آرایه G که توسط `find_colours` بازگردانده شده است، کاملاً با آرایه C یکسان باشد (یعنی $G[i] = C[i]$ برای همه i که $0 \leq i < N$). در غیر این صورت، شما 50% از امتیاز یک زیرمسئله را دریافت می‌کنید اگر شرایط زیر در همه تست کیس‌های آن برقرار باشد:

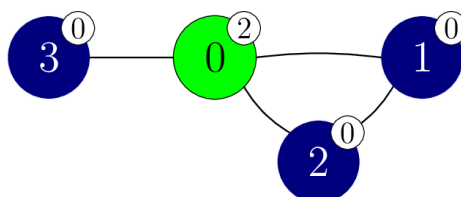
- $0 \leq G[i] < N$ برای هر i که $0 \leq i < N$;
- برای هر j که $0 \leq j < M$;
 $G[X[j]] = G[Y[j]]$ اگر و تنها اگر $C[X[j]] = C[Y[j]]$.

مثال

فرض کنید که فراخوانی زیر انجام شده است:

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

در این مثال، فرض کنید که رنگ‌های (پنهان) رأس‌ها به صورت زیر داده شده‌اند: $C = [2, 0, 0, 0]$. این سناریو در شکل زیر نشان داده شده است. رنگ‌ها همچنین با اعداد روی برچسب‌های سفید که به هر رأس متصل شده‌اند نشان داده شده‌اند.



این تابع ممکن است `perform_experiment` را به صورت زیر فراخوانی کند:

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

در این فراخوانی، هیچ رأسی دوباره رنگ‌آمیزی نمی‌شود و همه رأس‌ها رنگ اصلی خود را حفظ می‌کنند.

رأس 1 و رأس 2 را در نظر بگیرید. هر دوی آن‌ها رنگ 0 دارند و مسیر 1,2 یک مسیر تک‌رنگ است. در نتیجه، رأس‌های 1 و 2 در یک مولفه تک‌رنگ قرار دارند.

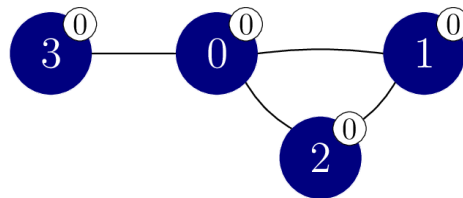
رأس 1 و رأس 3 را در نظر بگیرید. اگرچه هر دو رنگ 0 دارند، اما در مولفه‌های تک‌رنگ متفاوتی قرار دارند زیرا هیچ مسیری تک‌رنگ آن‌ها را به هم متصل نمی‌کند.

به طور کلی، 3 مولفه تک‌رنگ وجود دارد، با رأس‌های $\{0\}$ ، $\{1,2\}$ و $\{3\}$. بنابراین، این فراخوانی مقدار 3 را بازمی‌گرداند.

اکنون تابع ممکن است `perform_experiment` را به صورت زیر فراخوانی کند:

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

در این فراخوانی، فقط رأس 0 به رنگ 0 رنگ‌آمیزی می‌شود، که منجر به رنگ‌آمیزی نشان داده شده در شکل زیر می‌شود.

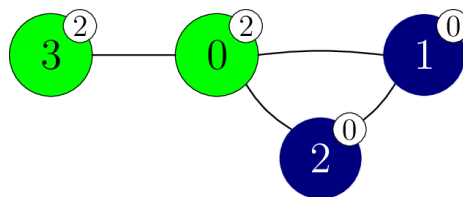


این فراخوانی مقدار 1 را بازمی‌گرداند، زیرا همه رأس‌ها در یک مولفه تک‌رنگ قرار دارند. اکنون می‌توانیم نتیجه بگیریم که رأس‌های 1، 2 و 3 رنگ 0 دارند.

سپس تابع ممکن است `perform_experiment` را به صورت زیر فراخوانی کند:

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

در این فراخوانی، رأس 3 به رنگ 2 رنگ‌آمیزی می‌شود، که منجر به رنگ‌آمیزی نشان داده شده در شکل زیر می‌شود.



این فراخوانی مقدار 2 را بازمی‌گرداند، زیرا 2 مولفه تک‌رنگ وجود دارد، با رأس‌های $\{0,3\}$ و $\{1,2\}$ به ترتیب. می‌توانیم نتیجه بگیریم که رأس 0 رنگ 2 دارد.

در نهایت، تابع `find_colours` آرایه $[2,0,0,0]$ را بازمی‌گرداند. از آنجا که $C = [2,0,0,0]$ ، امتیاز کامل داده می‌شود.

توجه داشته باشید که مقادیر بازگشتی دیگری نیز وجود دارند که 50% از امتیاز به آنها تعلق می‌گیرد، به عنوان مثال $[1, 2, 2, 2]$ یا $[1, 2, 2, 3]$.

ارزیاب نمونه

فرمت ورودی:

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

فرمت خروجی:

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

در اینجا، L طول آرایه G است که توسط `find_colours` بازگردانده شده است، و Q تعداد فراخوانی‌های انجام‌شده به `perform_experiment` است.