



سیمرغ

بنا بر افسانه‌ای قدیمی در شاهنامه، زال، قهرمان افسانه‌ای ایران به‌طور دیوانه‌واری دل‌داده رودابه شاهزاده کابل است. هنگامی‌که زال از رودابه درخواست ازدواج می‌کند، پدر رودابه چالشی را برای او طرح می‌کند.

در ایران، n شهر که از 0 تا $n - 1$ برچسب‌گذاری شده، و m جاده دوطرفه که از 0 تا $m - 1$ برچسب‌گذاری شده وجود دارد. هر جاده یک جفت شهر متفاوت را به هم متصل می‌کند. هر جفت شهر با حداکثر یک جاده به هم وصل شده‌اند. بعضی از جاده‌ها، جاده سلطنتی هستند که توسط افراد سلطنتی استفاده می‌شوند. وظیفه زال این است که مشخص کند کدام جاده‌ها سلطنتی هستند.

زال یک نقشه از همه شهرها و جاده‌های ایران دارد. او نمی‌داند که کدامیک از جاده‌ها سلطنتی هستند، اما می‌تواند از سیمرغ، پرنده پاک‌بین افسانه‌ای که محافظ زال است، کمک بگیرد. با این حال، سیمرغ نمی‌خواهد مجموعه جاده‌های سلطنتی را به‌طور مستقیم برای زال آشکار کند. به‌جای آن، به زال می‌گوید که مجموعه جاده‌های سلطنتی یک مجموعه طلایی است. یک مجموعه از جاده‌ها، یک مجموعه طلایی است اگر و تنها اگر دارای شرایط زیر باشد:

- دقیقاً شامل $n - 1$ جاده باشد، و
- برای هر جفت از شهرها، بتوان از شهر اول تنها با حرکت بر جاده‌های این مجموعه به شهر دوم رسید.

به علاوه، زال می‌تواند از سیمرغ سوال‌هایی بپرسد. برای هر سوال:

1. زال یک مجموعه طلایی از جاده‌ها را انتخاب می‌کند، و بعد
2. سیمرغ به زال می‌گوید که چندتا از جاده‌های مجموعه انتخاب شده توسط زال، جاده‌های سلطنتی هستند.

برنامه شما باید به زال کمک کند تا با پرسیدن حداکثر q سوال از سیمرغ، مجموعه جاده‌های سلطنتی را پیدا کند. ارزیاب نقش سیمرغ را بازی می‌کند.

جزئیات پیاده‌سازی

شما باید تابع زیر را پیاده‌سازی کنید:

```
int[] find_roads(int n, int[] u, int[] v)
```

- n : تعداد شهرها،
- u و v : آرایه‌هایی به طول m هستند. برای هر $0 \leq i \leq m - 1$ ، $u[i]$ و $v[i]$ شهرهایی هستند که با جاده i به هم متصل هستند.
- این تابع باید یک آرایه‌ای به طول $n - 1$ ، شامل برچسب جاده‌های سلطنتی را (به ترتیب دلخواه) برگرداند.

پاسخ شما می‌تواند حداکثر q فراخوانی از این تابع ارزیاب انجام دهد:

```
int count_common_roads(int[] r)
```

- r : آرایه‌ای به طول $n - 1$ شامل برچسب‌های جاده‌ها در یک مجموعه طلایی (به ترتیب دلخواه).
- این تابع تعداد جاده‌های سلطنتی در r را برمی‌گرداند.

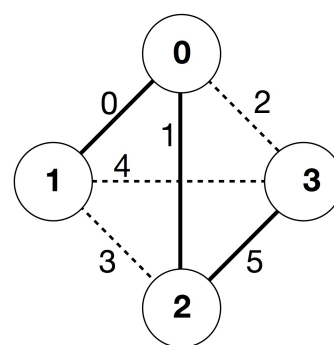
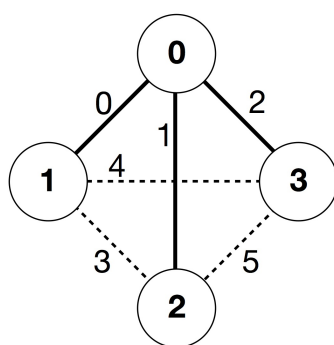
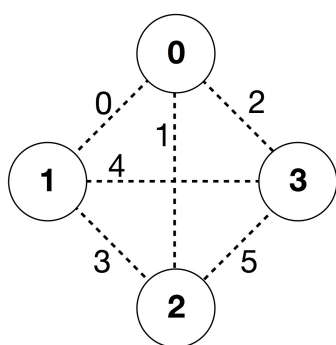
مثال

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

find_roads(...)

count_common_roads([0, 1, 2]) = 2

count_common_roads([5, 1, 0]) = 3



در این مثال ۴ شهر و ۶ جاده وجود دارد. ما یک جاده که دو شهر a و b را به هم متصل می‌کند را با (a, b) نشان می‌دهیم. جاده‌ها از ۰ تا ۵ به این ترتیب برچسب‌گذاری شده‌اند: $(0, 1)$ ، $(0, 2)$ ، $(0, 3)$ ، $(1, 2)$ ، $(1, 3)$ و $(2, 3)$. هر مجموعه طلایی $3 = n - 1$ جاده دارد.

فرض کنید جاده‌های سلطنتی جاده‌هایی هستند که با $0, 1$ ، و 5 برچسب‌گذاری شده‌اند، یعنی جاده‌های $(0, 1)$ ، $(0, 2)$ ، و $(2, 3)$. سپس:

- $\text{count_common_roads}([0, 1, 2])$ مقدار ۲ را برمی‌گرداند. این پرسش مربوط به جاده‌های برچسب‌گذاری شده با $0, 1$ ، و 2 یعنی جاده‌های $(0, 1)$ ، $(0, 2)$ ، و $(0, 3)$ است. دو تا از آن‌ها جاده‌های سلطنتی هستند.
- $\text{count_common_roads}([5, 1, 0])$ مقدار ۳ را برمی‌گرداند. این پرسش مربوط به مجموعه همه جاده‌های سلطنتی است.

تابع find_roads باید مقدار $[5, 1, 0]$ را برگرداند، یا هر آرایه‌ای به طول ۳ که شامل این سه عضو است.

توجه کنید که فراخوانی‌های زیر مجاز نیستند:

- $\text{count_common_roads}([0, 1])$: در اینجا طول r برابر با ۳ نیست.
- $\text{count_common_roads}([0, 1, 3])$: در اینجا r یک مجموعه طلایی را مشخص نمی‌کند، زیرا سفر از شهر ۰ به شهر ۳ تنها با استفاده از جاده‌های $(0, 1)$ ، $(0, 2)$ ، و $(1, 2)$ ممکن نیست.

محدودیت‌ها

- $2 \leq n \leq 500$
- $n - 1 \leq m \leq n(n - 1)/2$
- $0 \leq u[i], v[i] \leq n - 1$ (برای هر $0 \leq i \leq m - 1$)
- برای هر $0 \leq i \leq m - 1$ ، جاده i دو شهر متفاوت را به هم متصل می‌کند (یعنی، $u[i] \neq v[i]$).
- حداکثر یک جاده بین هر دو شهر وجود دارد.
- حرکت بین هر دو شهر از طریق جاده‌ها امکان‌پذیر است.
- مجموعه همه جاده‌های سلطنتی یک مجموعه طلایی است.
- `find_roads` باید تابع `count_common_roads` را حداکثر q بار فراخوانی کند. در هر فراخوانی مجموعه جاده‌های مشخص شده توسط r باید یک مجموعه طلایی باشد.

زیرمسئله‌ها

1. (۱۳ امتیاز) $q = 30\,000$, $n \leq 7$
2. (۱۷ امتیاز) $q = 30\,000$, $n \leq 50$
3. (۲۱ امتیاز) $q = 30\,000$, $n \leq 240$
4. (۱۹ امتیاز) $q = 12\,000$ و بین هر جفت شهر یک جاده وجود دارد
5. (۳۰ امتیاز) $q = 8000$

ارزیاب نمونه

ارزیاب نمونه ورودی را با قالب زیر می‌خواهند:

- سطر 1: n m
- سطر $2 + i$: $u[i]$ $v[i]$ (برای هر $0 \leq i \leq m - 1$)
- سطر $2 + m$: $s[0]$ $s[1]$... $s[n - 2]$

در اینجا، $s[0]$ ، $s[1]$ ، ...، $s[n - 2]$ برچسب‌های جاده‌های سلطنتی هستند.

ارزیاب نمونه مقدار YES را به عنوان خروجی می‌دهد، اگر `find_roads` تابع `count_common_roads` را حداکثر 30 000 بار فراخوانی کند، و مجموعه صحیح جاده‌های سلطنتی را برگرداند. در غیر این صورت، NO را به عنوان خروجی می‌دهد.

توجه کنید که تابع `count_common_roads` در ارزیاب نمونه بررسی نمی‌کند که r همه خواص مجموعه طلایی را داشته باشد. در مقابل، تعداد برچسب‌های جاده‌های سلطنتی در آرایه r را می‌شمارد و برمی‌گرداند. اما، اگر برنامه‌ای که شما ارسال می‌کنید تابع `count_common_roads` را با مجموعه‌ای از برچسب‌ها فراخوانی کند که یک مجموعه طلایی را توصیف نمی‌کند، نتیجه امتیازدهی 'Wrong Answer' خواهد بود.

نکات فنی

تابع `count_common_roads` در C++ و Pascal به دلایل کارایی از روش ارسال با ارجاع (pass by reference) استفاده می‌کند. شما همچنان می‌توانید این تابع را به روش متداول فراخوانی کنید. ارزیاب تضمین می‌کند که مقدار r را تغییر نخواهد داد.