Simurgh

Podle starověkých perských legend z Shahnamehu se Zal, legendární perský hrdina, šíleně zamiloval do Rudaby, princezny z Kábulu. Otec podmínil souhlas se sňatkem splněním zvláštního úkolu.

V Persii je n měst, označených od 0 do $n-1$, a m obousměrných silnic, číslovaných 0 až $m-1$. Každá silnice spojuje dvě různá města. Každá dvojice měst je spojena nejvýše jednou silnicí. Některé z těchto silnic jsou *královské silnice* používané pro cestování králů. Zalovým úkolem je určit, které silnice jsou královské silnice.

Zal má mapu se všemi městy a silnicemi v Persii. Neví, které ze silnic jsou královské, ale může získat pomoc od Simurgha, milostivého mýtického ptáka, který je Zalovým ochráncem. Simurgh mu ale množinu královských cest nechce odhalit přímo. Namísto toho mu řekne, že množina všech královských cest je *zlatou množinou*. Množina cest je zlatou množinou právě tehdy, když:

- Má *přesně* $n-1$ silnic a
- pro každou dvojici měst je možné docestovat z jednoho do druhého pouze po silnicích z této množiny.

Navíc se Zal může Simurgha ptát na tyto otázky. Pro každou otázku:

1. Zal vybere jednu *zlatou* množinu silnic a
2. Simurgh mu k ní odpoví, kolik cest ve zvolené zlaté množině jsou královské silnice.

Váš program by měl Zalovi pomoci najít množinu královských silnic tak, že se Simurgha zeptá na nejvýše q otázek. Vyhodnocovač bude hrát roli Simurgha.

Podrobnosti k implementaci

Máte za úkol implementovat následující proceduru:

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

- n: počet měst,
- u a v: jsou pole délky m. Pro všechna $0 \leq i \leq m-1$, jsou $u[i]$ a $v[i]$ města spojená silnicí i.
- Procedura musí vrátit pole délky $n-1$ obsahující čísla královských silnic (v libovolném pořadí).

Vaše řešení smí provést maximálně q volání následující procedury vyhodnocovače:

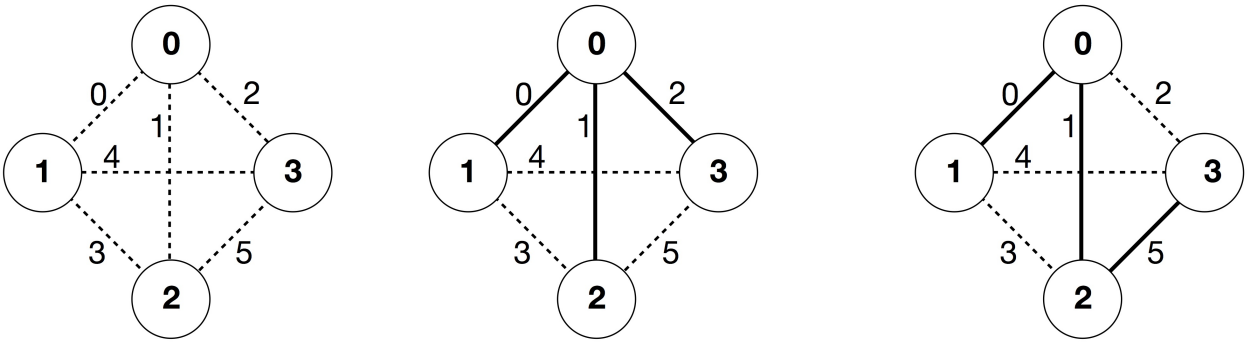
```
int count_common_roads(int[] r)
```

- r: je pole délky $n-1$ obsahující čísla silnic zlaté množiny (v libovolném pořadí).
- Procedura vrací počet královských silnic v r.

Příklad

```
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`



V tomto příkladu jsou 4 města a 6 silnic. Označme (a, b) silnici spojující města a a b. Silnice jsou číslovány 0 až 5 v následujícím pořadí: $(0, 1)$, $(0, 2)$, $(0, 3)$, $(1, 2)$, $(1, 3)$ a $(2, 3)$. Každá zlatá množina má $n - 1 = 3$ silnice.

Předpokládejme, že královskou množinu tvoří silnice číslované 0, 1 a 5, tj. silnice $(0, 1)$, $(0, 2)$ a $(2, 3)$. Pak platí:

- `count_common_roads([0, 1, 2])` vrátí 2. Tento dotaz se týká silnic číslovaných $0, 1$ a 2, tj. silnic $(0, 1)$, $(0, 2)$ a $(0, 3)$. Dvě z nich jsou královské.
- `count_common_roads([5, 1, 0])` vrátí 3. Tento dotaz se týká množiny všech královských silnic.

Procedura `find_roads` musí vrátit `[5, 1, 0]` nebo jakékoli jiné pole délky 3 obsahující tyto tři prvky.

Všimněte si, že následující volání nejsou přípustná:

- `count_common_roads([0, 1])`: zde délka r není 3.
- `count_common_roads([0, 1, 3])`: zde r nepopisuje zlatou množinu, protože z města 0 se do města 3 nelze po silnicích $(0, 1)$, $(0, 2)$, $(1, 2)$ dostat.

Omezení

- $2 \leq n \leq 500$

- $n - 1 \leq m \leq n(n-1)/2$
- $0 \leq u[i], v[i] \leq n - 1$ (pro všechna $0 \leq i \leq m - 1$)
- Pro všechna $0 \leq i \leq m - 1$ silnice i spojuje dvě různá města (tj. $u[i] \neq v[i]$).
- Mezi každými dvěma městy je nejvýše jedna silnice.
- Po silnicích je možné cestovat mezi libovolnou dvojicí měst.
- Množina všech královských cest je zlatou množinou.
- `find_roads` smí volat `count_common_roads` nejvýše q-krát. Při každém volání musí být množina silnic r zlatá.

Podúlohy

1. (13 bodů) $n \leq 7$, $q = 30\ 000$
2. (17 bodů) $n \leq 50$, $q = 30\ 000$
3. (21 bodů) $n \leq 240$, $q = 30\ 000$
4. (19 bodů) $q = 12\ 000$ a existuje silnice mezi všemi dvojicemi měst
5. (30 bodů) $q = 8000$

Ukázkový vyhodnocovač

Ukázkový vyhodnocovač čte vstup v následujícím formátu:

- řádek 1: n m
- řádek $2 + i$ (for all $0 \leq i \leq m - 1$): $u[i]$ $v[i]$
- řádek $2 + m$: $s[0]$ $s[1]$ $\ldots$ $s[n-2]$

Zde jsou $s[0], s[1], \ldots, s[n-2]$ čísla královských cest.

Ukázkový vyhodnocovač vypíše `YES`, jestliže `find_roads` zavolá `count_common_roads` nejvýše $30\ 000$-krát a vrátí správnou množinu královských silnic. Jinak vypíše `NO`.

Všimněte si, že procedura `count_common_roads` v ukázkovém vyhodnocovači neověřuje, že r má všechny vlastnosti zlaté množiny. Pouze vrátí počet královských cest v r. Platí však, že pokud váš program zavolá `count_common_roads` s množinou cest nikoli zlatou, bude řešení vyhodnoceno jako 'Wrong Answer'.

Technická poznámka

Procedura `count_common_roads` v C++ a Pascalu používá *předání hodnoty referencí* (pass by reference) z důvodu efektivity. Můžete nicméně proceduru volat i obvyklým způsobem. Máte zaručeno, že vyhodnocovač hodnotu r nezmění.