



Simurgh

Prema drevnoj Perzijskoj legendi iz Shahnamaha, Kile (legendarni Perzijski heroj) smrtno je zaljubljen u Paulu, princezu iz Kabula. Kada je Kile pitao Paulina oca za ruku njegove kćeri, otac mu je umjesto dozvole dao sljedeći task.

Perzija se sastoji od n gradova numeriranih od 0 do $n - 1$ i m dvosmjernih cesti numeriranih od 0 do $m - 1$. Svaka cesta spaja dva različita grada te je svaki par gradova direktno povezan s najviše jednom cestom. Neke su ceste *kraljevske* jer se po njima kreću kraljevi i te su ceste tajne. Kiletov je task odgonetnuti koje su ceste kraljevske.

Kile ima kartu svih gradova i svih cesti u Perziji. Ne zna koje su ceste kraljevske, ali može zatražiti pomoć od Pinte, mitološke spiritualne ptice *narančaste* boje koja je Kiletov zaštitnik. Nažalost, bezdušni Pinta ne želi samo tako Kiletu otkriti koje su ceste kraljevske. Pinta je Kiletu otkrio da kraljevske ceste čine *zlatni skup*. Skup cesta je zlatan ako i samo ako:

- sadrži točno $n - 1$ cestu i
- Svaki je par gradova povezan nekim podskupom cesta koje se nalaze u zlatnom skupu

Nadalje, Kile može pitati Pintu neka pitanja. Za svako pitanje:

1. Kile odabire *zlatan* skup cesti, a potom
2. Pinta odgovori Kiletu koliko je kraljevskih cesti u tom skupu.

Budući da Kile baš i ne zna programirati, vaš je zadatak pomoći mu tako da napišete program koji će postaviti najviše q pitanja Pinti. Za potrebe ovog zadatka, Grejder će igrati ulogu Pinte koji nije mogao biti onsite jer je pojeo gljivu.

Implementacijski detalji

Trebate implementirati sljedeće funkcije:

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

- n : broj gradova,
- u i v : nizovi duljine m . Za sve $0 \leq i \leq m - 1$, $u[i]$ i $v[i]$ su gradovi spojeni i -tom cestom.
- Ova funkcija vraća niz duljine $n - 1$ koji sadrži oznake kraljevskih cesti (u proizvoljnom poretku).

Vaše rješenje smije napraviti najviše q poziva sljedećoj grejderovoj funkciji:

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

- r : niz duljine $n - 1$ koji sadrži oznake cesti nekog zlatnog skupa (u proizvoljnom poretku).
- Ova funkcija vraća broj kraljevskih cesti u nizu r .

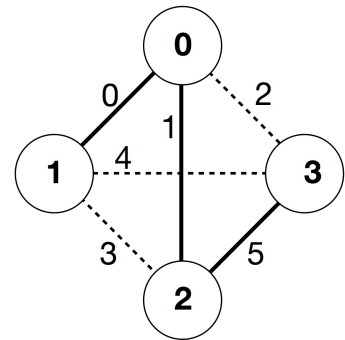
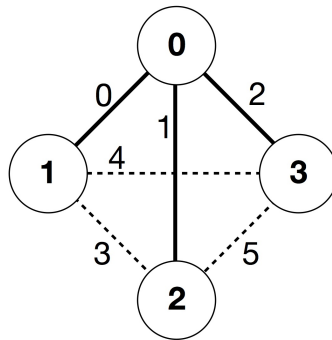
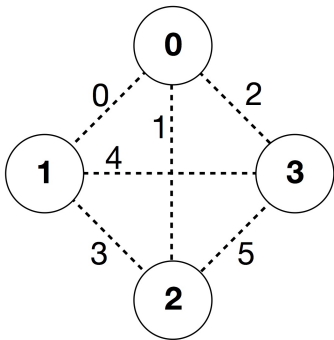
Primjer

```
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
```



U ovom primjeru Perzija se sastoji od 4 grada i 6 cesta. Sa (a, b) označavamo cestu koja spaja gradove a i b . Ceste su označene od 0 do 5 u sljedećem poretku: $(0, 1)$, $(0, 2)$, $(0, 3)$, $(1, 2)$, $(1, 3)$, i $(2, 3)$. Svaki zlatni skup ima $n - 1 = 3$ ceste.

Pretpostavite da su ceste s oznakama 0, 1 i 5, odnosno $(0, 1)$, $(0, 2)$, i $(2, 3)$, kraljevske i da program postavlja sljedeće upite:

- `count_common_roads([0, 1, 2])` vraća 2. Ovaj je upit o cestama s oznakama 0, 1, i 2, odnosno cestama $(0, 1)$, $(0, 2)$ i $(0, 3)$. Dvije od njih su kraljevske ceste.
- `count_common_roads([5, 1, 0])` vraća 3. Ovaj upit sadrži sve kraljevske ceste.

Funkcija `find_roads` trebala bi vratiti `[5, 1, 0]` ili bilo koji drugi niz duljine 3 koji sadrži ova tri elementa.

Primijetite da sljedeći pozivi nisu dozvoljeni:

- `count_common_roads([0, 1])`: ovdje duljina niza r nije 3.
- `count_common_roads([0, 1, 3])`: ovdje r ne opisuje zlatni skup jer nije moguće proputovati od grada 0 do grada 3 koristeći ceste $(0, 1)$, $(0, 2)$ i $(1, 2)$.

Ograničenja

- $2 \leq n \leq 500$
- $n - 1 \leq m \leq n(n - 1)/2$

- $0 \leq u[i], v[i] \leq n - 1$ (za sve $0 \leq i \leq m - 1$)
- Za sve $0 \leq i \leq m - 1$, cesta i spaja dva različita grada (dakle, $u[i] \neq v[i]$).
- Postoji najviše jedna cesta između svakog para gradova.
- Moguće je proputovati između bilo koja dva grada koristeći ceste iz ulaza.
- Skup svih kraljevskih cesta je zlatni skup.
- `find_roads` smije pozvati `count_common_roads` najviše q puta. U svakom pozivu skup cesti sadržan u nizu r mora biti zlatni skup.

Podzadaci

1. (13 bodova) $n \leq 7, q = 30\,000$
2. (17 bodova) $n \leq 50, q = 30\,000$
3. (21 bodova) $n \leq 240, q = 30\,000$
4. (19 bodova) $q = 12000$ i postoji cesta između svakog para gradova.
5. (30 bodova) $q = 8000$

Sempl grejder

Sempl grejder čita ulaz u sljedećem formatu:

- redak 1: $n \ m$
- redak $2 + i$ (za sve $0 \leq i \leq m - 1$): $u[i] \ v[i]$
- redak $2 + m$: $s[0] \ s[1] \ \dots \ s[n - 2]$

Ovdje su $s[0], s[1], \dots, s[n - 2]$ oznake kraljevskih cesta.

Sempl grejder vraća YES, ako `find_roads` zove `count_common_roads` najviše 30 000 puta i vraća točan kraljevski skup. U protivnom, ispisuje NO.

Uzmite u obzir da `count_common_roads` u sempl grejderu ne provjerava ima li r sva svojstva zlatnog skupa. Ta funkcija broji i vraća koliko se oznaka kraljevskih cesta nalazi u nizu r . Naravno, ako program koji submitate poziva `count_common_roads` sa skupom cesta koje nisu zlatni skup, dobit ćete 'Wrong Answer'.

Tehnička napomena

Funkcija `count_common_roads` u C++ i Pascalu koristi *call by reference* zbog efikasnosti. I dalje možete pozvati funkciju na klasičan način. Garantirano je da grejder neće mijenjati niz r .