Simurgh

U staroj persijskoj legendi u Shahnameh-u, legendarni heroj Zal je bio ludo zaljubljen u Rudabu, princezu Kabula. Kad je Zal zaprosio Rudabu, kako to već biva u legendama, njen otac je odlučio da mu postavi zadatak koji Zal mora riješiti da bi se vjenčao sa svojom dragom.

Postoji n gradova u Persiji, numerisanih brojevima od 0 do $n-1$, i m dvosmjernih puteva, označenih brojevima od 0 do $m-1$. Svaki put povezuje par različitih gradova, a svaka dva grada su povezana najviše jednim putem. Neki od puteva su *kraljevski*, jer ih koriste samo članovi kraljevske porodice i ti putevi su državna tajna. Zalov zadatak je da odredi koji su od puteva *kraljevski*.

Zal ima mapu svih gradova i puteva u Persiji. Nije mu poznato koji su putevi kraljevski, ali ima pomoć bezbrižne mitske ptice Simurgh, koja je po legendi Zalov zaštitnik. Simurgh ne može Zalu direktno reći koji su putevi *kraljevski*. Umjesto toga, ona govori Zalu da je skup *kraljevskih* puteva *zlatni skup*. Skup puteva je *zlatan skup* ako i samo ako su ispunjeni sljedeći uslovi:

1. skup ima tačno $n-1$ put
2. za svaka dva grada moguće je doći od jednog do drugog grada putujući samo putevima iz datog skupa.

Pored toga Zal može Simurgh-u postavljati neka pitanja. Za svako pitanje:

1. Zal bira *zlatni skup* puteva, i zatim
2. Simurgh daje odgovor Zalu koliko ima *kraljevskih* puteva u tom *zlatnom skupu*.

Vaš program treba da pomogne Zalu da pronađe *kraljevske puteve* postavljajući najviše q pitanja Simurgh-u.

Detalji implementacije

Implementirajte sljedeću funkciju/proceduru:

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

- n: broj gradova,
- u i v: nizovi dužine m. Za svako $0 \leq i \leq m-1$, $u[i]$ i $v[i]$ su gradovi koje povezuje put i.
- Ova funkcija/procedura vraća niz dužine $n-1$ koji sadrži oznake kraljevskih puteva (u proizvoljnom poretku).

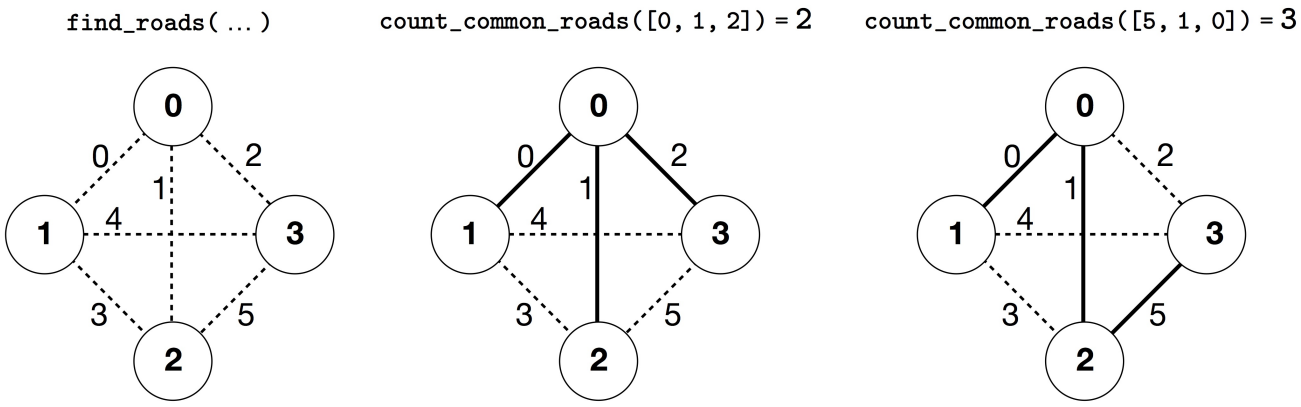
Vaše rješenje može najviše q puta pozvati sljedeće funkcije grejdera:

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

- r: niz dužine $n-1$ koji sadrži oznake puteva u *zlatnom skupu* (u proizvoljnom poretku).
- Ova funkcija vraća broj *kraljevskih puteva* u skupu predstavljenim nizom r.

Primjer

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



U ovom primjeru postoje 4 grada i 6 puteva. Označimo sa (a, b) put koje povezuje gradove a i b. Putevi su numerisani od 0 do 5 u sljedećem poretku: $(0, 1)$, $(0, 2)$, $(0, 3)$, $(1, 2)$, $(1, 3)$ i $(2, 3)$. Svaki *zlatni skup* ima $n - 1 = 3$ puta.

Pretpostavimo da su putevi 0, 1 i 5 (tj. putevi $(0, 1)$, $(0, 2)$ i $(2, 3)$) *kraljevski putevi*, i da program izvršava sljedeće pozive:

- `count_common_roads([0, 1, 2])` vraća 2. Ovo pitanje je o putevima čije su oznake 0, 1 i 2 (tj. putevi $(0, 1)$, $(0, 2)$ i $(0, 3)$). Dva od njih su *kraljevski putevi*.
- `count_common_roads([0, 1, 2])` vraća 3. Ovo je pitanje o skupu svih *kraljevskih puteva*.

Funkcija `find_roads` treba da vrati $[5, 1, 0]$ ili bilo koji drugi niz dužine 3 koji sadrži ta tri elementa.

Obratite pažnju da sljedeći pozivi nisu dopušteni:

- `count_common_roads([0,1 ])`: dužina niza r nije 3.
- `count_common_roads([0, 1, 3])`: u ovom slučaju niz r ne opisuje *zlatni skup*, jer nije moguće doći iz grada 0 do grada 3 koristeći samo puteve $(0, 1)$, $(0, 2)$ 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 svako $0 \leq i \leq m - 1$)
- Za svako $0 \leq i \leq m - 1$, put i povezuje dva različita grada (tj. $u[i] \neq v[i]$).
- Postoji najviše jedan put između bio koja dva grada.
- Moguće je putovati između bilo koja dva grada koristeći date puteve.
- Skup svih kraljevskih puteva jeste zlatan skup.
- `find_roads` može pozivati funkciju `find_common_roads` najviše q puta. Pri svakom pozivu, skup opisan nizom r mora biti zlatan skup.

Podzadaci

1. (13 bodova) $n \leq 7$, $q = 30\,000$
2. (17 bodova) $n \leq 50$, $q = 30\,000$
3. (21 bod) $n \leq 240$, $q = 30\,000$
4. (19 bodova) $q = 12\,000$ i postoji put između svaka dva grada
5. (30 bodova) $q = 8000$

Primjer grejdera (programa za ocjenjivanje)

Dati grejder (program za ocjenjivanje) učitava podatke u sljedećem formatu:

- red 1: n m
- red $2 + i$ (za svako $0 \leq i \leq m - 1$): $u[i]$ $v[i]$
- red $2 + m$: $s[0]$ $s[1]$ $\ldots$ $s[n - 2]$

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

Grejder vraća validan skup kraljevskih puteva i daje izlaz `YES`, ako funkcija `find_roads` poziva `count_common_roads` najviše $30\,000$ puta . U suprotnom, grejder daje izlaz `NO`.

Obratite pažnju da `count_common_roads` u datom grejderu ne provjerava da li r opisuje zlatni skup. Umjesto toga, samo prebrojava koliko ima kraljevskih puteva u nizu r i vraća broj takvih puteva. Međutim, ako vaš program poziva `count_common_roads` sa skupom oznaka koji ne opisuju zlatni skup, CMS će vam vratiti 'Wrong Answer'.

Tehnička napomena

Funkcija `count_common_roads` u C++ i Pascal-u, radi efikasnosti, koristi *prenošenje parametara po referenci* (engl. *call by reference*). Vi je možete pozivati na uobičajeni način. Garantuje se da grejder neće mijenjati vrijednosti u nizu r.