



# Simurgh

Po starih perzijskih pripovedkah iz Sahnameha, je legendarni perzijski princ Zal do ušes zatreskan v princeso iz Kabula, Rudabo. Ko je Zal zaprosil Rubadinega očeta za njeno roko, mu je on postavil izziv.

V Perziji je  $n$  cest, označenih s števili od 0 do  $n - 1$  in  $m$  dvosmernih cest, označenih s števili od 0 do  $m - 1$ . Vsaka cesta povezuje dve različni mest. Vsak par mest je povezan z največ eno cesto. Nekatere ceste so *kraljeve ceste*, ki so namenjene potovanju kraljeve družine. Zalova naoga je, da določi, katere ceste so kraljeve.

Zal ima zemljevid vseh mest in vseh cest v Perziji. Ne ve pa, katere ceste so kraljeve. Pomaga mu lahko Zalova zavetnica Simurgh, mitološka ptica. Vendar Simurgh noče uničiti naloge in neposredno razkriti vseh kraljevih cest. Namesto tega mu pove, da je množica kraljevih cest *zlata množica*. Množica cest je zlata množica če in samo če velja:

- Ima *natanko*  $n - 1$  cest in
- za vsak par mest je možno priti od enega do drugega le po cestah iz te množice.

Nadalje lahko Zal vpraša Simurgh nekaj vprašanj. Za vsako vprašanje velja:

1. Zal izbere *zlato* množico cest in nato
2. Simurgh pove Zalu, koliko cest v zlati množici je kraljevih.

Tvoj program naj pomaga Zalu odkriti množico kraljevih cest, s spraševanjem Simurgh. Vpraša lahko največ  $q$  vprašanj. Ocenjevalnik bo v tej igri igral vlogo Simurgh.

## Podrobnosti implementacije

Implementirati moraš naslednjo funkcijo:

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

- $n$ : število mest,
- $u$  in  $v$ : polji dolžine  $m$ . Za vse  $0 \leq i \leq m - 1$ ,  $u[i]$  in  $v[i]$  predstavljata mesti, povezani s cesto  $i$ .
- Funkcija naj vrne polje dolžine  $n - 1$ , ki vsebuje oznake kraljevih cest (v katerem koli zaporedju).

Tvoja rešitev lahko naredi največ  $q$  klicev ocenjevalnikove funkcije:

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

- $r$ : polje dolžine  $n - 1$ , ki vsebuje oznake cest zlata množice (v katerem koli zaporedju).
- Funkcija vrne število kraljevih cest v  $r$ .

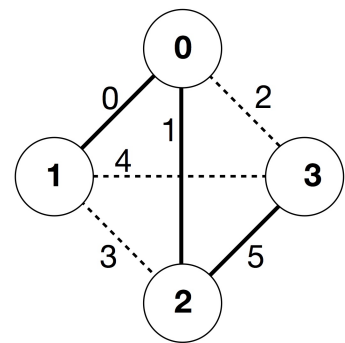
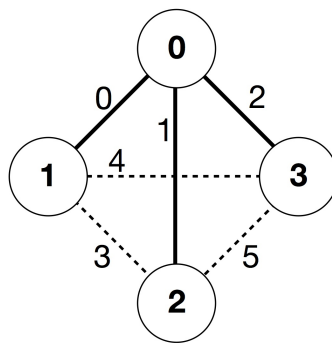
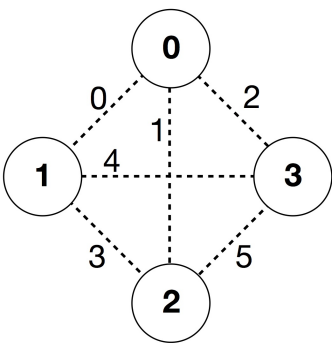
## Primer

```
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 tem primeru so 4 mesta in 6 cest. Cesto, ki povezuje mesti  $a$  in  $b$ , označimo z  $(a, b)$ . Ceste so označene s števili od 0 do 5 v naslednjem vrstnem redu:  $(0, 1)$ ,  $(0, 2)$ ,  $(0, 3)$ ,  $(1, 2)$ ,  $(1, 3)$  in  $(2, 3)$ . Vsaka zlata množica ima  $n - 1 = 3$  cest.

Denimo, da so kraljeve ceste označene z 0, 1 in 5, torej ceste  $(0, 1)$ ,  $(0, 2)$  in  $(2, 3)$ . Torej:

- `count_common_roads([0, 1, 2])` vrne 2. Poizvedba je narejena na cestah označenih z 0, 1 in 2, torej cestah  $(0, 1)$ ,  $(0, 2)$  in  $(0, 3)$ . Dve od njih sta kraljevi.
- `count_common_roads([5, 1, 0])` vrne 3. Ta poizvedba je množica vseh kraljevih cest.

Funkcija `find_roads` naj vrne `[5, 1, 0]` ali kakršno koli polje dolžine 3, ki vsebuje iste elemente.

Vedi, da naslednji klici niso dovoljeni:

- `count_common_roads([0, 1])`: tukaj dolžina  $r$  ni enaka 3.
- `count_common_roads([0, 1, 3])`: tukaj  $r$  ne opisuje zlata množice, ker iz mesta 0 ni mogoče potovati do mesta 3, saj so v množici le ceste  $(0, 1)$ ,  $(0, 2)$ ,  $(1, 2)$ .

## Omejitve

- $2 \leq n \leq 500$
- $n - 1 \leq m \leq n(n - 1)/2$
- $0 \leq u[i], v[i] \leq n - 1$  (za vse  $0 \leq i \leq m - 1$ )
- Za vse  $0 \leq i \leq m - 1$ , cesta  $i$  povezuje dve različni mesti (npr.  $u[i] \neq v[i]$ ).
- Med vsakim parom mest obstaja največ ena cesta.
- Med poljubnim parom mest je možno potovati po cestah.
- Množica vseh kraljevih cest je zlata množica.
- `find_roads` naj kliče `count_common_roads` največ  $q$  krat. Pri vsakem klicu naj bo množica cest v  $r$  zlata množica.

## Podnaloge

1. (13 točk)  $n \leq 7, q = 30\,000$
2. (17 točk)  $n \leq 50, q = 30\,000$
3. (21 točk)  $n \leq 240, q = 30\,000$
4. (19 točk)  $q = 12\,000$  in med vsakim parom mest obstaja cesta.
5. (30 točk)  $q = 8000$

## Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod naslednjega formata:

- 1. vrstica:  $n \ m$
- $2 + i$ -ta vrstica (za vse  $0 \leq i \leq m - 1$ ):  $u[i] \ v[i]$
- $2 + m$ -ta vrstica:  $s[0] \ s[1] \ \dots \ s[n - 2]$

Tukaj so  $s[0], s[1], \dots, s[n - 2]$  oznake kraljevih cest.

Vzorčni ocenjevalnik izpiše YES, če funkcija `find_roads` kliče funkcijo `count_common_roads` največ 30 000 krat in vrne pravilno množico kraljevih cest. Sicer izpiše NO.

Funkcija `count_common_roads` v vzorčnem ocenjevalniku ne preverja, če  $r$  ustreza lastnostim zlate množice. Namesto tega prešteje in vrne število oznak kraljevih cest v polju  $r$ . Če bo program klical funkcijo `count_common_roads` z množico, ki ne ustreza pogojem zlate množice, bo tekmovalni ocenjevalnik vrnil napako 'Wrong Answer'.

## Tehnična opomba

Funkcija `count_common_roads` v programskima jeziku C++ in Pascal uporablja metodo *pass by reference* zaradi učinkovitosti. Še vedno lahko kličeš funkcijo normalno. Zagotovljeno je, da ocenevalnik ne bo spreminjal vrednosti  $r$ .