



# Simurga

Saskaņā ar seno persiešu poētisko eposu Šāhname, leģendārais varonis Zāls bija traki iemīļījis Kabulas princesē Rudabā. Kad Zāls lūdza Rudabas roku, viņas tēvs uzdeva Zālam sarežģītu uzdevumu.

Persijā ir  $n$  pilsētas, kas sanumurētas no 0 līdz  $n - 1$ , un  $m$  divvirzienu ceļi, kas sanumurēti no 0 līdz  $m - 1$ . Katrs ceļš savieno divas atšķirīgas pilsētas. Katru pilsētu pāri savieno ne vairāk kā viens ceļš. Daži ceļi, kurus ceļojumiem izmanto augstmaņi, ir *karaliski*, un to numuri tiek glabāti noslēpumā. Zāla uzdevums ir noteikt, kuri no ceļiem ir karaliski.

Zālam ir visu Persijas pilsētu un ceļu karte. Viņš nezina, kuri ceļi ir karaliski, bet viņš var lūgt palīdzību Simurgai – devīgajam mītiskajam putnam, kas ir Zāla aizstāvis. Tomēr Simurga nevēlas tieši atklāt karalisko ceļu kopu. Tā vietā Simurga ir pateikusi Zālam, ka karaliskie ceļi veido *zelta kopu*. Ceļu kopa is zelta kopa tad un tikai tad, ja:

- tajā ir tieši *tieši*  $n - 1$  ceļš, un
- katram pilsētu pārim ir spēkā īpašība, ka no vienas pilsētas ir iespējams aizbraukt uz otru, izmantojot tikai šīs kopas ceļus.

Turklāt Zāls var uzdot Simurgai noteikta veida jautājumus. Katram jautājumam:

1. Zāls izvēlas kādu *zelta* kopu, un tad
2. Simurga Zālam atbild, cik šīs zelta kopas ceļi ir karaliski.

Jūsu programmai jāpalīdz Zālam atrast karalisko ceļu kopu, uzdodot Simurgai ne vairāk kā  $q$  jautājumu. Vērtētājs spēlēs Simurgas lomu.

## Realizācijas noteikumi

Jums jārealizē procedūra

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

- $n$ : pilsētu skaits,
- $u$  un  $v$ : masīvi garumā  $m$ . Visiem  $i$  ( $0 \leq i \leq m - 1$ ),  $u[i]$  un  $v[i]$  ir pilsētas, kuras savieno ceļš  $i$ .
- Šai procedūrai jāatgriež masīvs garumā  $n - 1$ , kas satur visus karaliskos ceļus patvaļīgā secībā.

Jūsu risinājums ne vairāk kā  $q$  reizes drīkst izsaukt procedūru

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

- $r$ : masīvs garumā  $n - 1$ , kas satur kādas zelta kopas ceļu numurus patvaļīgā secībā.
- Procedūra atgriež karalisko ceļu skaitu kopā  $r$ .

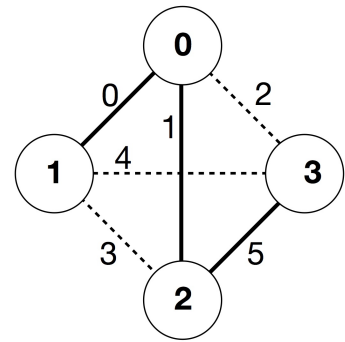
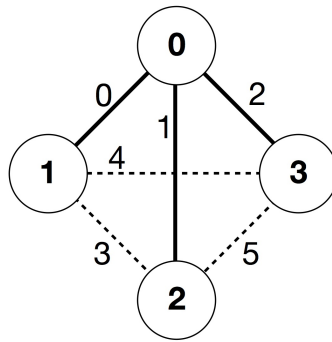
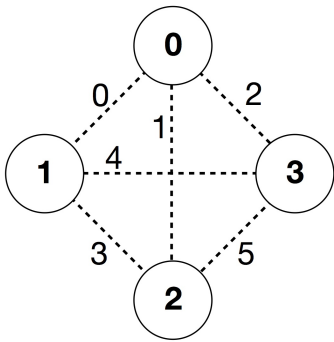
## Piemērs

```
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
```



Šajā piemērā ir 4 pilsētas un 6 ceļi. Ar  $(a, b)$  tiek apzīmēts ceļš, kas savieno pilsētas  $a$  un  $b$ . Ceļi ir numurēti no 0 līdz 5 šādā secībā:  $(0, 1)$ ,  $(0, 2)$ ,  $(0, 3)$ ,  $(1, 2)$ ,  $(1, 3)$ , un  $(2, 3)$ . Katrā zelta kopā ir  $n - 1 = 3$  ceļi.

Pieņemsim, ka karaliskie ceļi ir ceļi ar numuriem 0, 1, un 5, tas ir, ceļi  $(0, 1)$ ,  $(0, 2)$ , un  $(2, 3)$ . Tad

- `count_common_roads([0, 1, 2])` atgriež skaitli 2. Šis vaicājums ir par ceļiem ar iezīmēm 0, 1, un 2, tas ir, ceļiem  $(0, 1)$ ,  $(0, 2)$  un  $(0, 3)$ . Divi no šiem ceļiem ir karaliski.
- `count_common_roads([5, 1, 0])` atgriež skaitli 3. Šis vaicājums ir par visu karalisko ceļu kopu.

Procedūrai `find_roads` jāizvada  $[5, 1, 0]$  vai jebkurš cits masīvs garumā 3, kas satur šos trīs elementus.

Ievērojiet, ka šādi izsaukumi nav atļauti:

- `count_common_roads([0, 1])`: masīva  $r$  garums nav 3.
- `count_common_roads([0, 1, 3])`: masīvs  $r$  neapraksta zelta kopu, jo nav iespējams no pilsētas 0 nokļūt pilsētā 3, izmantojot tikai ceļus  $(0, 1)$ ,  $(0, 2)$  un  $(1, 2)$ .

## Ierobežojumi

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

- $0 \leq u[i], v[i] \leq n - 1$  visiem  $i$  ( $0 \leq i \leq m - 1$ )
- Visiem  $i$  ( $0 \leq i \leq m - 1$ )  $i$ -tais ceļš savieno divas atšķirīgas pilsētas (t.i.,  $u[i] \neq v[i]$ ).
- Starp katrām divām pilsētām ir ne vairāk kā viens ceļš.
- No jebkuras pilsētas iespējams nokļūt jebkurā citā, izmantojot ceļus.
- Karalisko ceļu kopa ir zelta kopa.
- `find_roads` drīkst izsaukt `count_common_roads` ne vairāk kā  $q$  reizes. Katrā izsaukumā masīvā  $r$  uzdoto ceļu kopai jābūt zelta kopai.

## Apakšuzdevumi

1. (13 punkti)  $n \leq 7, q = 30\,000$
2. (17 punkti)  $n \leq 50, q = 30\,000$
3. (21 punkts)  $n \leq 240, q = 30\,000$
4. (19 punkti)  $q = 12\,000$  un eksistē ceļš starp katrām divām pilsētām
5. (30 punkti)  $q = 8000$

## Paraugvērtētājs

Paraugvērtētājs lasa ievaddatus šādā formātā:

- 1. rinda:  $n \ m$
- $(2 + i)$ -tā rinda (visiem  $i$  ( $0 \leq i \leq m - 1$ )):  $u[i] \ v[i]$
- $(2 + m)$ -tā rinda:  $s[0] \ s[1] \ \dots \ s[n - 2]$

$s[0], s[1], \dots, s[n - 2]$  ir karalisko ceļu numuri.

Paraugvērtētājs izvada YES, ja `find_roads` izsauc `count_common_roads` ne vairāk kā 30 000 reizi un atgriež pareizu karalisko ceļu kopu. Pretējā gadījumā tas izvada NO.

Esiet uzmanīgi, jo paraugvērtētāja procedūra `count_common_roads` **nepārbauda**, vai masīvā  $r$  ir aprakstīta zelta kopa. Tā vietā tā saskaita un atgriež karalisko ceļu numuru skaitu masīvā  $r$ . Tomēr, ja jūsu iesūtītā programma izsauks `count_common_roads` ar masīvu, kas neapraksta zelta kopu, vērtējums būs 'Wrong Answer'.

## Tehniska piezīme

Procedūra `count_common_roads` valodās C++ un Pascal efektivitātes dēļ izmanto *vērtības nodošanu pēc references*. Jūs joprojām varat izsaukt procedūru ierastajā veidā. Ir garantēts, ka vērtētājs nemainīs  $r$  vērtību.