



Simurgh

Legendaarne Pärsia kangelane Zal on kõrvuni armunud Kabuli printsessi Rudabasse. Selleks, et saada õigust Rudaba kätt küsida, peab Zal lahendama tema isa poolt püstitatud väljakutse.

Pärsias on n linna, mis on nummerdatud 0 kuni $n - 1$. Linnad on omavahel ühendatud m kahe-suunalise tee abil. Teed on nummerdatud 0 kuni $m - 1$. Iga tee ühendab kaks erinevat linna, ning iga kahe linna vahel on ülimalt üks tee. Mõned teed on *kuninglikud* - neid kasutavad oma liikumise jaoks kuningliku perekonna liikmed. Seda, millised teed on *kuninglikud*, hoitakse tavainimeste eest saladuses. Zal'i ülesandeks ongi välja selgitada, millised teed on *kuninglikud*.

Zal'il on Pärsia kõigi linnade ja teede kaart. Ta ise ei tea, millised on kuninglikud teed, kuid talle pakub abi tema kaitsja - müütiline lind Simurgh. Kahjuks ei taha Simurgh paljastada kuninglikkude teede hulka otse. Selle asemel andis ta Zal'ile teada, et kõik kuninglikud teed moodustavad *kuldse hulga* ning lubas teatud kujul küsimusi esitada.

Teede hulk on *kuldne* parajasti siis, kui:

- hulgas on täpselt $n-1$ teed, ning
- igast linnast saab igasse teise linna jõuda kasutades ainult selle hulga teid.

Zal võib esitada Simurghile küsimusi järgnevalt:

1. Zal mõtleb välja mõne *kuldse* teede hulga, ning
2. Simurgh annab teada, mitu teed valitud hulgas on *kuninglikud*.

Sinu programm peab aitama Zal'il välja selgitada kogu kuninglike teede hulga, esitades kuni q küsimust (kus q on alamülesande-spetsiifiline piirang).

Tehnilised detailid

Lahendusena tuleb realiseerida järgmine protseduur:

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

- n : linnade arv,
- u ja v : massiivid pikkusega m . Iga $0 \leq i \leq m - 1$ puhul on linnad $u[i]$ ja $v[i]$ ühendatud teega i .
- Protseuur peab tagastama massiivi pikkusega $n - 1$, mis loetleb kuninglikkude teede numbrid (suvalises järjekorras).

Lahendus võib kutsuda välja järgmist testris defineeritud protseduuri:

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

- r : massiiv pikkusega $n - 1$, milles on antud ühe *kuldse* teede hulga numbrid suvalises järjekorras.
- Funktsioon tagastab hulgas r esinevate kuninglikkude teede arvu.

Funktsiooni võib välja kutsuda kuni q korda (kus q sõltub alamülesandest). Selle piirangu ületamisel saab lahendus vastavast testist 0 punkti.

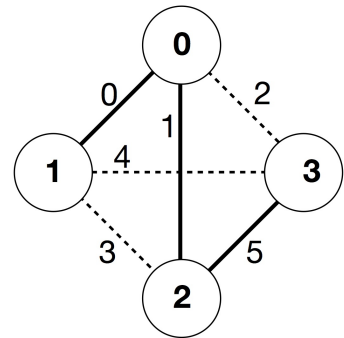
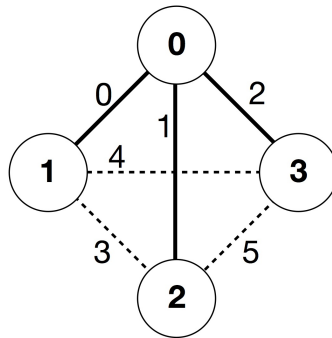
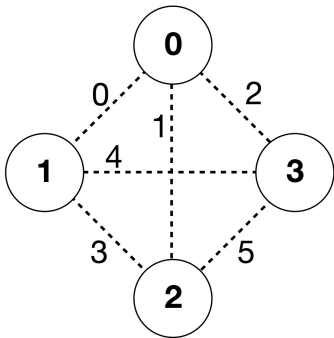
Näide

```
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`



Antud näites on 4 linna ja 6 teed. Linnade a ja b vahel asuvat teed tähistame kui (a, b) . Linnadevahelised teed (nummerdatud 0 kuni 5) on seega järgmised: $(0, 1)$, $(0, 2)$, $(0, 3)$, $(1, 2)$, $(1, 3)$, ja $(2, 3)$. Iga kuldne hulk peab koosnema $n - 1 = 3$ teest.

Olgu kuninglikud teed 0, 1, ja 5, s.t. $(0, 1)$, $(0, 2)$, and $(2, 3)$. Oletame et programm teeb järgmised väljakutsed:

- `count_common_roads([0, 1, 2])` tagastab 2. Siin päritakse infot teedest $(0, 1)$, $(0, 2)$ ja $(0, 3)$ koosneva kuldse hulga kohta. Kaks nendest teedest on kuninglikud.
- `count_common_roads([5, 1, 0])` tagastab 3. Selles päringus on juba loetletud parajasti kõik kuninglikud teed.

Funktsioon `find_roads` peab tagastama massiivi $[5, 1, 0]$ (või nende kolme arvu suvalise ümberpaigutuse).

Pane tähele, et järgmised väljakutsed pole lubatud:

- `count_common_roads([0, 1])`: siin pole massiivi r pikkus 3.
- `count_common_roads([0, 1, 3])`: siin ei kirjelda massiiv r kuldset teede hulka, kuna

näiteks linnast 0 pole võimalik jõuda linna 3 kasutades ainult teid (0, 1), (0, 2), (1, 2).

Piirangud

- $2 \leq n \leq 500$
- $n - 1 \leq m \leq n(n - 1)/2$
- $0 \leq u[i], v[i] \leq n - 1$ (iga $0 \leq i \leq m - 1$ jaoks)
- Iga $0 \leq i \leq m - 1$ jaoks, tee i ühendab kaks erinevat linna (s.t., $u[i] \neq v[i]$).
- Iga kahe linna vahel on ülimalt üks tee.
- Iga linnast võib teede abil igasse teise linna jõuda.
- Kõigi kuninglike teede hulk on *kuldne*.
- `find_roads` võib funktsiooni `count_common_roads` välja kutsuda maksimaalselt q korda. Igas väljakutses peab massiivis r antud teede hulk olema *kuldne*.

Alamülesanded

1. (13 punkti) $n \leq 7$, $q = 30\,000$
2. (17 punkti) $n \leq 50$, $q = 30\,000$
3. (21 punkti) $n \leq 240$, $q = 30\,000$
4. (19 punkti) $q = 12\,000$ ning iga kahe linna vahel on tee
5. (30 punkti) $q = 8000$

Näidistester

Näidistester loeb sisendit järgmises formaadis:

- rida 1: n m
- rida $2 + i$ (iga $0 \leq i \leq m - 1$ jaoks): $u[i]$ $v[i]$
- rida $2 + m$: $s[0]$ $s[1]$ \dots $s[n - 2]$

Siin, $s[0], s[1], \dots, s[n - 2]$ on kuninglike teede numbrid.

Näidistester väljastab YES, kui `find_roads` kutsub protseduuri `count_common_roads` ülimalt 30 000 korda, ning tagastab korrektse kuninglike teede hulga. Muidu väljastab ta NO.

Pange tähele, et näidistestris antud `count_common_roads` funktsioon *ei kontrolli*, et r rahuldaks kõik kuldse hulga tingimusi. Ta lihtsalt loeb kokku ja tagastab kuninglike teede arvu massiivis r . Samas, kui esitatud lahenduses kutsutakse protseduuri `count_common_roads` sisendiga, mis ei kirjelda kuldset teede hulka, on hindamistulemuseks 'Wrong Answer'.

Märkus

Funktsiooni `count_common_roads` realisatsioon C++ ja Pascalis võtab parameetri vastu viitena ainult efektiivsuse jaoks. Tegelikult tester ei muuda massiivi r sisu.