



Simurg'

Shohnomadagi qadimiy fors afsonalariga ko'ra afsonaviy fors qahramoni Zal Qolib malikasi Rudobani telbalarcha sevadi. Zal Rudobani qo'lini so'raganda qizning otasi yigitga jumboq berdi.

Persiyada n ta shahar bor va ular 0 dan $n - 1$ gacha belgilangan va m ta 2 yo'nalishli yo'l bo'lib, ular 0 dan $m - 1$ gacha belgilangan. Har bir yo'l bir juft aniq shaharni bog'lab turadi. Har juft shahar ko'pi bilan bitta yo'l oqali bog'lanadi. yo'llarning ayrimlari qirollik yo'lari bo'lib bu yo'llardan qurollik oila a'zolari yurishadi. Zalni vazifasi yo'llardan qay bir qirol oilasiga tegishli ekanligini aniqlash, yani qurollik yo'llari tizimini.

Zalda Presiyadagi barcha shaharlar va yo'llarni haritasi bor. U qaysi yo'l qirolni yo'li ekanligini bilmaydi. Lekin, afsonaviy mehribon qush Simurghdan yordam olish mumkin. Chunki bu qush Zalning himoyachisidir. Shunga qaramay Simurgh qirollik yo'llari sirini to'g'ridan to'g'ri oshkor etmoqchi emas. Buning o'rniga u qirollik yo'llarining tizimi ham oltin tizim ekani aytdi.

Oltin tizim quyidagi shartlarni qanoatlantiradi:

- u aynan $n - 1$ yo'llarga ega, va
- har bir juft shaharga faqat shu yo'l orqaligina yetib bo'rish mumkin.

Aniqrog'i oltin tizim bu daraxt desa ham bo'ladi. Shuningdek, Zal Simurghdan bir necha savollar so'rashi mumkin. Har bir savol uchun:

1. Zal yo'llar 'oltin' tizimi tanlaydi
2. Simurgh Zalga tallangan oltin tizimdan nechtasi qirollik yo'llar tizimga tegishli yo'l borligini aytadi.

Sizning dastingiz Zalga Simurghga ko'pi bilan q ta savol berish orqali qirollik yo'llar tizimini topishda yordam berish kerak. Graderga savollar jo'natiladi.

Implementation details

Siz qo'llashigiz kerak bo'lgan funksiya ko'rinish:

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

- n :shaharlar soni,
- u va v : uzunligi m bo'lgan massiv. $0 \leq i \leq m - 1$ uchun $u[i]$ va $v[i]$ ni bog'lovchi i yo'l.
- Funksiya $n - 1$ uzunlikdagi massiv qaytish kerak. Qirollik yo'llari qaytariladi(yo'llar ixtiyoriy tartibda).

Sizning funksiyagiz ko'pi bilan q marta murojat qilish kerak:

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

- $r : n - 1$ uzunlikdagi massiv oltin tizimdagi yo'llarni ifodalaydi(ihtiyoriy tartibda).
- Funksiya qirollik yo'llar soni r massivda nechtaligi qaytariladi.

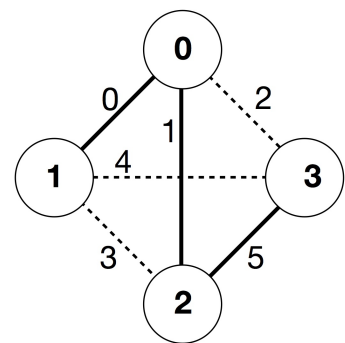
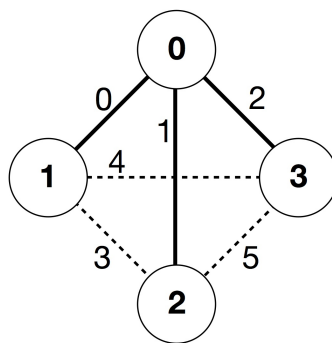
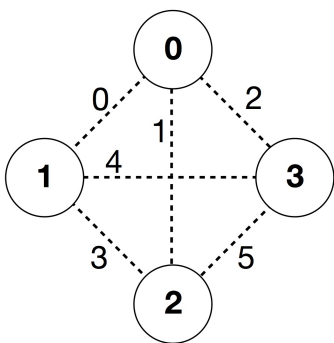
Example

```
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`



Bu namunada 4 shahar va 6 ta yo'l bor. Biz a va b shaharni bog'lovchi yo'lni (a, b). 0 dan 5 gacha yo'llar quyidagi tartibda beriladi: (0, 1), (0, 2), (0, 3), (1, 2), (1, 3) va (2, 3). Oltin tizim $n - 1 = 3$ teng..

Tasvur qiling 0, 1 va 5 bilan belgilangan yo'llar qirollik yo'llari ifodalangan yo'llar (0,1), (0, 2) va (2, 3), keyin:

- `count_common_roads([0, 1, 2])` natija 2. 0, 1 va 2 berilganda bular ichida 2 ta qirollik yo'li mavjud.
- `count_common_roads([5, 1, 0])` natija 3. hammasi qirollik yo'llari ekan.

Funksiyagiz `find_roads` natija sifatida [5, 1, 0] qaytarish kerak yoki shularning kombinatsiyasi bo'lish mumkin.

Esda tuting quyidagi so'rovnomalarni berish ruhsat etilmaydi:

- `count_common_roads([0, 1])` : bu yerda r ning uzunligi 3 emas.
- `count_common_roads([0, 1, 3])` : bu yerda r oltin tizimini tasvirlamaydi. Chunki 0 shaharda 3 shaharga faqatgina (0, 1), (0, 2), (1, 2) yo'llar bo'ylab yetib borishning iloji yo'q.

Constraints

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

- $0 \leq u[i], v[i] \leq n - 1$
- $u[i] \neq v[i]$
- har juft shahar o'rtasida ko'pi bilan bitta yo'l bor.
- Hohlagan juft shahar o'rtasida yo'llar orqali yurish mumkin.
- Qirollik yo'li tizmi oltin tizim hisoblanadi.
- `find_roads` `count_common_roads`ga ko'pi bilan q marta so'rov berishi mumkin. Har bir so'rovda r oltin tizim bo'lish kerak.

Subtasks

1. (13 points) $n \leq 7, q = 30\,000$
2. (17 points) $n \leq 50, q = 30\,000$
3. (21 points) $n \leq 240, q = 30\,000$
4. (19 points) $q = 12000$ va har bir juft shahar o'rtasida yo'l bor
5. (30 points) $8000 \leq q \leq 30\,000$

Sample grader

Graderning o'qish farmati:

- 1 qator: $n\ m$
- $2 + i$ qator: $u[i], v[i]$
- $2 + m$: $s[0], s[1], \dots, s[n - 2]$

bu yerda $s[0], s[1], \dots, s[n - 2]$ qirollik yo'larining ifodalari.

Namuna grader YES chiqari berish uchun `find_roads` ga `count_common_roads` ga ko'pi bilan 30000 marta so'rov bersa va to'g'ri qirollik yo'li qaytarsa. Aks holda NO chiqaradi. Habardor bo'ling namuna graderdagi `count_common_roads` r massiv oltin tizimning barcha hosalariga egami yo'qmiligini tekshirmaydi. Buning o'rniga u r massivdagi qirollik yo'llari ifodalari soni sanaydi va natija qilib qaytaradi. Shunga qaramay siz taqdim qilayotgan dastur oltin tizimni tasvirlamaydigan ifodalarga so'rov bersa 'Wrong Answer' natija qaytardi. .

Technical note

The procedure `count_common_roads` in C++ and Pascal uses the *call by reference* method for efficiency reasons. You can still call the procedure in the usual way. The grader is guaranteed not to change the value of r .