



Симург

Эртний Персийн нэгэн домогт өгүүлсэнээр Персийн домогт баатар Зал Кабулын гүнж Рудабад ухаангүй дурласан ажээ. Зал Рудабатай гэрлэх хүсэлт тавих үед гүнжийн аав түүнд нэгэн даалгавар өгчээ.

Перс улсад 0-ээс $n - 1$ хүртэл дугаарласан n хот, мөн 0-ээс $m - 1$ дугаартай m зам байдаг. Зам бүр хоёр ялгаатай хотыг холбодог ба аль ч хоёр хот хамгийн ихдээ нэг замаар холбогдсон байна. Зарим замыг *сурвалжитны зам* гэх ба түүгээр сурвалжит гаралтай хүмүүс аялдаг. Иймээс эдгээр замууд нууцлагдсан байдаг. Залын даалгавар аль замууд *сурвалжит зам* болохыг тогтоох явдал юм. Залд Перс дэх хот болон замыг агуулсан газрын зураг өгөгдсөн. Тэр аль зам сурвалжит эсэхийг мэдэхгүй боловч өөрийн хамгаалагч болох сайхан сэтгэлт шувуу Симургээс тусламж авч чадна. Гэвч Симург аль зам сурвалжит болохыг шууд хэлж өгөхийг хүсээгүй бөгөөд үүний оронд тэрээр бүх сурвалжит замын олонлог нь *алтан олонлог* болохыг хэлжээ. Зөвхөн дараах нөхцөл биелэх үед ямар нэг замын олонлогийг *алтан олонлог* гэнэ. Үүнд:

- Уг олонлог нь яг $n - 1$ ширхэг замаас тогтсон байх
- Ямар ч хоёр хотын хувьд нэгээс нөгөө рүү зөвхөн энэ олонлогт орсон замаар хүрч болдог байх

Үүнээс гадна Зал Симургээс асуулт асууж болно. Асуулт бүр дараах хэлбэртэй:

1. Зал замын алтан олонлог сонгоно
2. Симург сонгосон алтан олонлогийн замын хэд нь сурвалжит болохыг Залд хэлнэ

Таны програм Симургээс хамгийн олондоо q ширхэг асуулт асуун сурвалжит замын олонлогийг олж, Залд туслах ёстой. Грэйдер нь Симургийн дүрд тоглоно.

Implementation details

Дараах процедурыг бичнэ үү:

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

- n : хотын тоо
- u ба v : m урттай массивууд. Бүх $0 \leq i \leq m - 1$ хувьд $u[i]$ ба $v[i]$ нь i дугаар замаар холбогдсон хотууд.
- Энэ процедур сурвалжит замуудын дугаарыг агуулсан $n - 1$ урттай массив буцаах ёстой. Үүнд эрэмбэ хамаарахгүй.

Таны шийдэл дараах грэйдер процедурыг хамгийн ихдээ q удаа дуудаж болно:

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

- r : Ямар нэг алтан олонлог дахь замуудын дугаарыг агуулсан $n - 1$ урттай массив (эрэмбэ хамаарахгүй)
- Энэ процедур r дэх сурвалжит замын тоог буцаана

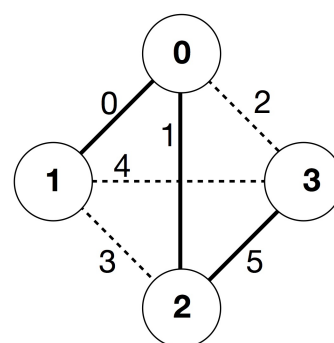
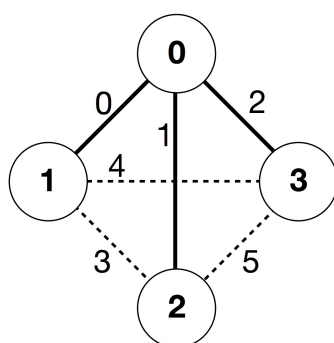
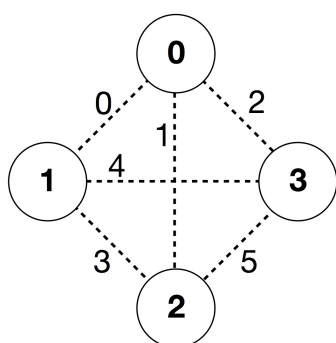
Example

```
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`



Энэ жишээнд 4 хот, 6 зам байна. (a, b) -аар a ба b хотыг холбосон замыг тэмдэглэе. Замууд дараах дарааллаар 0-ээс 5 хүртэл дугаарлагдсан байна: $(0, 1)$, $(0, 2)$, $(0, 3)$, $(1, 2)$, $(1, 3)$ болон $(2, 3)$. Алтан олонлог бүр $n - 1 = 3$ замтай байна.

0, 1 ба 5 дугаартай $(0, 1)$, $(0, 2)$ болон $(2, 3)$ замууд нь сурвалжит ба програм дараах асуулт асуусан байг. Үүнд:

- `count_common_roads([0, 1, 2])` нь 2 утга буцаасан. 0, 1, ба 2 дугаартай $(0, 1)$, $(0, 2)$ болон $(0, 3)$ замуудаас тогтох алтан олонлогийн хувьд асуулт асуусан байна. Эдгээр замын хоёр нь сурвалжит зам.
- `count_common_roads([5, 1, 0])` нь 3 утга буцаасан. Бүх сурвалжит замын хувьд асуусан байна.

`find_roads` процедур нь $[5, 1, 0]$ эсвэл эдгээр гурван дугаарыг агуулсан 3 урттай дурын массив буцаах ёстой.

Дараах асуултууд асуухыг зөвшөөрөхгүй болохыг анхаарна уу:

- `count_common_roads([0, 1])`: r -ийн урт 3 биш.
- `count_common_roads([0, 1, 3])`: r нь алтан олонлог биш. Учир нь 0 хотоос 3 хот руу $(0, 1)$, $(0, 2)$, $(1, 2)$ замуудаар явж хүрч чадахгүй.

Constraints

- $2 \leq n \leq 500$
- $n - 1 \leq m \leq n(n - 1)/2$
- $0 \leq u[i], v[i] \leq n - 1$ ($0 \leq i \leq m - 1$)
- Бүх $0 \leq i \leq m - 1$ хувьд i зам хоёр ялгаатай хотыг холбоно (ө.х. $u[i] \neq v[i]$).
- Ямар ч хоёр хотын хооронд хамгийн ихдээ нэг зам байна.
- Аль ч хоёр хотын хооронд замуудаар явж хүрэх боломжтой.
- Бүх сурвалжит замын олонлог нь алтан олонлог байна.
- `find_roads` нь `count_common_roads`-ийг хамгийн ихдээ q удаа дуудаж болно. Асуулт бүрт r -аар тодорхойлогдох замуудын олонлог алтан олонлог байх ёстой.

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) $q = 12000$ ба аль ч хоёр хот замаар холбогдсон
5. (30 points) $8000 \leq q \leq 30\,000$

Sample grader

Sample grader дараах форматаар өгөгдлийг уншина:

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

Энд $s[0], s[1], \dots, s[n - 2]$ нь сурвалжит замын дугаар.

Хэрэв `find_roads` нь зөв сурвалжит замын олонлог буцаасан ба `count_common_roads`-ийг хамгийн ихдээ q удаа дуудсан бол YES, үгүй бол NO гэсэн хариуг grader хэвлэнэ.

Sample grader дэх `count_common_roads` процедур r нь алтан олонлог эсэхийг бүрэн шалгахгүй гэдгийг санана уу. Үүний оронд энэ процедур r дэх сурвалжит замыг тоолж буцаадаг. Гэвч таны програм `count_common_roads`-ийг алтан олонлог биш байх дугаартай олонлогоор дуудвал 'Wrong Answer' хариу авна.

Technical note

C++ ба Pascal хэлэнд `count_common_roads` процедур *pass by reference* хэлбэр ашигласан байгаа. Та процедурыг ердийн байдлаар дуудаж болно. The grader нь r -ийн утгыг хэзээ ч өөрчлөхгүй.