

Станц (stations)

Сингапурын Интернетийн Гол Сүлжээ (Singapore's Internet Backbone - SIB) нь n тооны станцаас тогтох ба тэдгээрт 0 to $n - 1$ хүртлэх **индексүүд** оноосон байдаг. Мөн 0 -ээс $n - 2$ хүртлэх тоонуудаар дугаарласан $n - 1$ хоёр чиглэлтэй холболтууд байдаг. Холболт бүр нь хоёр ялгаатай станцыг холбоно. Нэг холболтоор холбогдсон хоёр станцыг хөрш гэж нэрлэнэ.

x станцаас y станц хүрэх зам гэдэг нь $a_0 = x$, $a_p = y$ ба дараалсан хоёр станц бүр нь хөрш байх a_0, a_1, \dots, a_p гэсэн ялгаатай станцуудын дараалал юм. Аль ч x станцаас өөр ямар ч y станц хүртэл **яг нэг** зам оршин байна.

Дурын x станц нь багц (хэсэг өгөгдөл) үүсгэн багцийн **хүлээн авагч** гэж нэрлэгдэх өөр ямар ч y станц руу илгээж болно. Уг багцыг x -ээс y хүрэх цорын ганц замаар доорх байдлаар дамжуулна.

Хүлээн авагч нь y байх пакетыг одоо эзэмшиж байгаа z станцыг авч үзье. ($z \neq y$). Уг станц нь:

- z -ээс y хүрэх цор ганц зам дээр байрлах z -ийн хөршийг тодорхойлох **чиглүүлэлтийн функц**-ийг ажиллуулж,
- пакетийг уг хөрш рүү дамжуулна.

Гэвч станцуудын санах ой нь хязгаарлагдмал тул чиглүүлэлтийн функцэд зориулан SIB дэх холболтуудын жагсаалтыг бүтнээр нь хадгалж чаддаггүй.

Та хоёр функцээс тогтох SIB-ийн чиглүүлэлтийн схемийг хэрэгжүүлэх болно.

- Эхний функцийн оролтон дээр нь n , SIB дэх холболтуудын жагсаалт, $k \geq n - 1$ бүхэл тоо өгөгдөнө. Уг функц нь станц бүрт 0 -ээс k хүртлэх бүхэл тоонуудаас давтагдашгүй **дугаар** олгоно.
- Хоёр дахь функц нь чиглүүлэлтийн функц ба дугаар олгогдсоны дараа бүх станц дээр хэрэглэгдэнэ. Уг функцэд **зөвхөн** доорх оролтуудыг өгнө:
 - s нь пакетыг эзэмшиж байгаа станцын **дугаар**,
 - t нь пакетын хүлээн авагч станцийн **дугаар** ($t \neq s$),
 - c нь s -ийн бүх хөршүүдийн **дугааруудын** жагсаалт.

Уг функц нь пакетын дараагийн эзэмшигч болох s -ийн хөршийн **дугаарыг** буцаана.

Мөн таны бодолтын оноо нь станцуудад оноосон дугаарын хамгийн их утгаас хамаарна (ерөнхийдөө бага бол сайн).

Хэрэгжүүлэлтийн мэдээлэл

Та дараах функцуудыг хэрэгжүүлнэ:

```
int[] label(int n, int k, int[] u, int[] v)
```

- n : SIB дэх станцуудын тоо.
- k : хэрэглэж болох дугаарын хамгийн их утга.
- u ба v : холболтуудыг илэрхийлэх $n - 1$ хэмжээтэй массивууд. i ($0 \leq i \leq n - 2$) утга бүрийн хувьд i холболт нь $u[i]$ болон $v[i]$ индекстэй станцуудыг холбоно.
- Уг функц нь n хэмжээтэй L гэсэн ганц массивыг буцаана. i ($0 \leq i \leq n - 1$) утга бүрийн хувьд $L[i]$ нь i индекстэй станцид оноосон дугаар байна. L массивын бүх элементүүд давтагдашгүй байх ба утга нь 0-ээс k хүртлэх тоонуудын нэг байна.

```
int find_next_station(int s, int t, int[] c)
```

- s : пакетийг эзэмшиж байгаа станцийн дугаар.
- t : пакетийн хүлээн авагч станцийн дугаар.
- c : s -ийн бүх хөршийн дугааруудын жагсаалтыг агуулах массив. c массив нь өсөхөөр эрэмбэлэгдсэн байна.
- Уг функц нь пакетын дараагийн эзэмшигч болох s -ийн хөршийн дугаарыг буцаана.

Тест бүр нь нэг эсвэл хэд хэдэн үзэгдлийг (өөрөөр хэлбэл ялгаатай SIB-ийн тодорхойлолтыг) агуулна. r үзэгдэл бүхий тестийн хувьд дээрх функцуудыг дуудаж байгаа **програмыг** доорх байдлаар яг хоёр удаа ажиллуулна.

Програмыг эхний удаа ажиллуулахад:

- `label` функцийг r удаа дуудна,
- буцаасан дугааруудыг шалгах систем хадгална,
- `find_next_station` функцийг дуудахгүй.

Програмыг хоёр дахь удаа ажиллуулахад:

- `find_next_station` функцийг олон удаа дуудаж болно,
- `find_next_station` функцийг дуудах бүрт програмыг эхний удаа ажиллуулах үеийн үзэгдлүүдээс аль нэгийг нь **санамсаргүйгээр сонгон** авч `label` функцийг ажиллуулахад үүсгэсэн дугааруудыг оролт болгон өгнө,
- `label` функцийг дуудахгүй.

Мөн, програмыг эхний удаа ажиллуулахад статик эсвэл глобал хувьсагчдад хадгалсан ямар ч мэдээлэл `find_next_station` функц дотроос харагдахгүй.

Жишээ

Доорх дуудалтыг авч үзье:

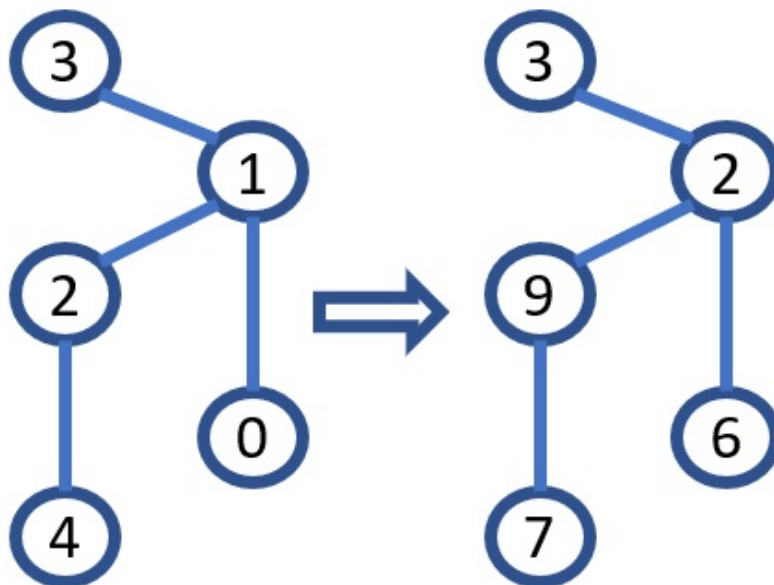
```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

Нийт 5 станц байгаа ба (0,1), (1,2), (1,3) болон (2,4) гэсэн хос станцуудыг холбосон 4 холболт байна. Дугаар бүр нь 0-ээс $k = 10$ хүртлэх бүхэл тоо байна.

Доорх дугаарлалтыг мэдэгдэхийн тулд:

Index	Label
0	6
1	2
2	9
3	3
4	7

label функц нь [6, 2, 9, 3, 7]-ийг буцаана. Доорх зурган дээр станц бүрийн хувьд зүүн гар талд индексүүдийг, баруун гар талд дугааруудыг харуулсан.



Дээр үзүүлсэн байдлаар дугаарлалтыг хийсэн гэж үзээд доорх дуудалтыг авч үзье:

```
find_next_station(9, 6, [2, 7])
```

Энэ нь 9 дугаартай станц пакетийг эзэмшиж байгаа ба хүлээн авагч станц нь 6 дугаартай байна гэсэн үг юм. Хүлээн авагч станц хүрэх зам дээр байгаа станцуудын дугаар нь [9, 2, 6] байна. Иймд уг дуудалт нь 2-г буцаана. Энэ бол пакетийн эзэмшигч болох дараагийн станцийн дугаар юм (индекс нь 1 байна).

Өөр нэг боломжит дуудалтыг авч үзье:

```
find_next_station(2, 3, [3, 6, 9])
```

Уг функц нь 3-ыг буцаана. Учир нь 3 дугаартай хүлээн авагч станц нь 2 дугаартай станцын хөрш тул пакетыг шууд хүлээн авах болно.

Хязгаарлалт

- $1 \leq r \leq 10$

label-ийн дуудалт бүрийн хувьд:

- $2 \leq n \leq 1000$
- $k \geq n - 1$
- $0 \leq u[i], v[i] \leq n - 1$ (бүх $0 \leq i \leq n - 2$ утгын хувьд)

find_next_station функцийн дуудалт бүрийн хувьд оролтыг нь өмнө хийгдсэн label-ийн дуудалтуудаас санамсаргүйгээр сонгон авна. Уг дуудалтаар дугааруудыг гарган авсан гэж үзье. Тэгвэл:

- s ба t нь ялгаатай станцуудын дугаарууд.
- c нь s дугаартай станцийн хөршүүдийн дугааруудын дараалал (өсөх эрэмбээр).

Бүх үзэгдлийн хувьд find_next_station функц руу дамжуулсан бүх c массивуудын нийт урт нь 100 000-аас хэтрэхгүй.

Дэд бодлого

1. (5 оноо) $k = 1000$, ямар ч станцын хөршийн тоо 2-оос хэтрэхгүй.
2. (8 оноо) $k = 1000$, i холболт нь $i + 1$ болон $\lfloor \frac{i}{2} \rfloor$ станцуудыг холбоно.
3. (16 оноо) $k = 1\,000\,000$, дээд тал нь нэг станц 2-оос олон хөрштэй байна.
4. (10 оноо) $n \leq 8$, $k = 10^9$
5. (61 оноо) $k = 10^9$

Дэд бодлого 5 дээр та хэсэгчилсэн оноо авч болно. Бүх үзэгдлийн хувьд m нь label-ийн буцаасан утгуудын хамгийн их нь байг. Энэ дэд бодлого дээр таны авах оноо нь доорх хүснэгтийн дагуу бодогдоно:

Максимум дугаар	Оноо
$m \geq 10^9$	0
$2000 \leq m < 10^9$	$50 \cdot \log_{5 \cdot 10^5}(\frac{10^9}{m})$
$1000 < m < 2000$	50
$m \leq 1000$	61

Жишээ шалгагч

Жишээ шалгагч нь оролтыг дараах хэлбэрээр уншина:

- мөр 1: r

Дараа нь тус бүр нь нэг үзэгдлийг тодорхойлох r ширхэг блок байрлана. Блок бүр нь доорх форматтай байна:

- мөр 1: $n \ k$
- мөр $2 + i$ ($0 \leq i \leq n - 2$): $u[i] \ v[i]$
- мөр $1 + n$: find_next_station-ын дуудалтын тоо, q .
- мөр $2 + n + j$ ($0 \leq j \leq q - 1$): $z[j] \ y[j] \ w[j]$: find_next_station функцийг j дэх удаа дуудахад оролцсон станцуудын **индексүүд**: $z[j]$ станц пакетийг эзэмшиж байгаа, $y[j]$ станц нь пакетийн хүлээн авагч станц, $w[j]$ станц нь $z[j]$ -ээс $y[j]$ хүрэх цорын ганц зам дээрх, пакетийн дараагийн эзэмшигч болох станц.

Жишээ шалгагч нь гаралтыг дараах хэлбэрээр гаргана:

- мөр 1: m

Дараа нь оролт дээрх үзэгдлүүдэд харгалзах r ширхэг блок байрлана. Блок бүр нь доорх форматтай байна:

- line $1 + j$ ($0 \leq j \leq q - 1$): уг үзэгдэл дээр find_next_station функцийг j дэх удаа дуудахад **дугаарыг** нь буцаасан станцийн **индекс**.

Жишээ шалгагч ажиллах бүртээ label болон find_next_station функцийг хоёуланг нь дуудаж байгааг анхааралдаа аваарай.