Stotelės (stations)

„Singapore's Internet Backbone“ (SIB) sudarytas iš n stotelių, kurioms priskirti **indeksai** nuo 0 iki $n-1$. Taip pat yra $n-1$ dvikrypčių jungčių, sunumeruotų nuo 0 iki $n-2$. Kiekviena jungtis sujungia 2 skirtingas stoteles. Dvi jungtimi sujungtos stotelės yra vadinamos kaimynėmis.

Kelias nuo stotelės x iki stotelės y yra skirtingų stotelių seka $a_0, a_1, \cdots, a_p$, kurioje $a_0 = x$, $a_p = y$, ir kiekvienos dvi gretimos stotelės kelyje yra kaimynės. Yra **lygiai vienas** kelias nuo bet kurios stotelės x iki bet kurios kitos stotelės y.

Bet kuri stotelė x gali sukurti duomenų paketą ir pasiųsti jį bet kuriai kitai stotelei y, kuri vadinama paketo **paskirties stotele**. Šis paketas turi nukeliauti vieninteliu keliu nuo x iki y tokiu būdu: tarkime, kad paketas yra stotelėje z, o to paketo paskirties stotelė yra y ($z \neq y$).

Šiuo atveju stotelė z:

1. atlieka **maršrutizavimo procedūrą**, kuri nustato z kaimynę, esančią veninteliame kelyje nuo z iki y, ir
2. persiunčia paketą savo kaimynei.

Tačiau stotelės turi ribotą atminties kiekį ir nesaugo visų SIB jungčių.

Parašykite SIB maršrutizavimo sistemą, kurią sudaro dvi funkcijos.

- Pirmoji funkcija gauna n, SIB jungčių sąrašą ir sveikąjį skaičių $k \geq n-1$. Ji kiekvienai stotelei priskiria **unikalią žymę**, kuri yra sveikasis skaičius nuo 0 iki k imtinai.
- Antroji funkcija atlieka maršrutizavimo procedūrą ir yra įdiegiama į visas stoteles po to, kai stotelėms priskiriamos žymės. Jai prieinami **tik** šie duomenys:
  - s, stotelės, kurioje dabar yra paketas, **žymė**,
  - t, paketo paskirties stotelės **žymė** ($t \neq s$),
  - c, visų s kaimynių **žymių** sąrašas.

  Ji turėtų grąžinti s kaimynės, kuriai turėtų būti perduotas paketas, **žymę**.

Vienoje iš dalinių užduočių jūsų gautas taškų kiekis priklausys nuo didžiausios priskirtos žymės vertės (kuo mažesnė didžiausia vertė, tuo geriau).

Realizacija

Jums reikia parašyti šias funkcijas:

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

- n: SIB stotelių skaičius.
- k: maksimali žymės vertė, kurią galima naudoti.
- u ir v: $n-1$ dydžio masyvai, nusakantys jungtis. Kiekvienam i ($0 \leq i \leq n-2$), jungtis i sujungia stoteles su indeksais $u[i]$ ir $v[i]$.
- Ši funkcija turėtų grąžinti vieną n dydžio masyvą L. Kiekvienam i ($0 \leq i \leq n-1$) $L[i]$ nurodo žymę, priskirtą stotelei su indeksu i. Visi masyvo L elementai privalo būti skirtingi sveikieji skaičiai nuo 0 iki k imtinai.

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

- s: stotelės, kurioje yra paketas, žymė.
- t: paketo paskirties stotelės žymė.
- c: masyvas, nurodantis visų s kaimynių žymių sąrašą. Masyvas c yra išrikiuotas didėjimo tvarka.
- Ši funkcija turėtų grąžinti s kaimynės, kuriai turėtų būti perduotas paketas, žymę.

Kiekvieną testą sudaro vienas arba daugiau tarpusavyje nepriklausomų scenarijų (t. y. skirtingų SIB aprašymų). Testui, kurį sudaro r scenarijų, **programa**, iškviečianti aukščiau apibūdintas funkcijas, yra paleidžiama lygiai du kartus žemiau nurodytu būdu.

Pirmuoju programos paleidimu:

- funkcija `label` iškviečiama r kartų,
- grąžintos žymės yra išsaugomos vertinimo sistemoje, ir
- funkcija `find_next_station` nėra iškviečiama.

Antruoju programos paleidimu:

- Funkcija `find_next_station` gali būti iškviesta keletą kartų. Kiekvieno iškvietimo metu **atsitiktinai** parenkamas scenarijus ir žymės, grąžintos tame scenarijuje funkcijos `label`, perduodamos funkcijai `find_next_station` kaip parametrai.
- Funkcija `label` nėra iškviečiama.

Atkreipkite dėmesį, kad jokia informacija, išsaugota statiniuose arba globaliuose kintamuosiuose paleidus programą pirmą kartą, nėra pasiekiama funkcijai `find_next_station`.

Pavyzdys

Panagrinėkime tokį iškvietimą:

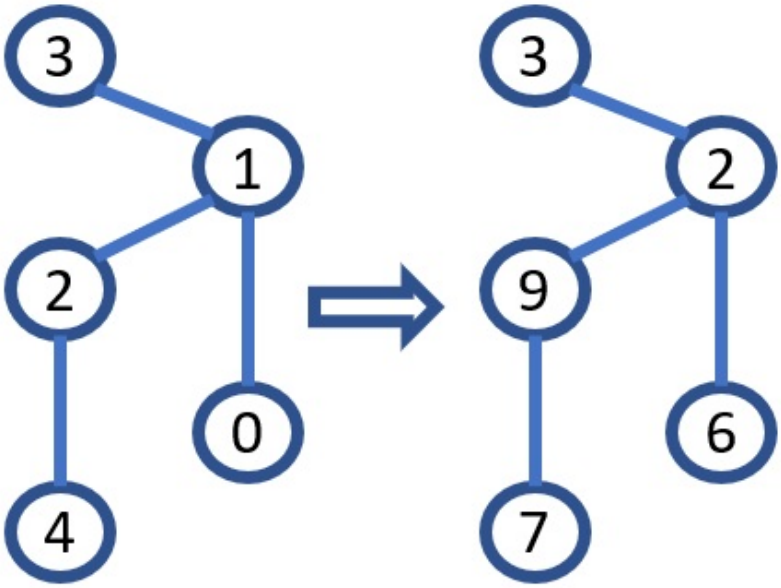
```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

Iš viso yra 5 stotelės ir 4 jungtys, jungiančios stoteles su indeksais $(0, 1)$, $(1, 2)$, $(1, 3)$ ir $(2, 4)$. Kiekviena žymė gali būti sveikasis skaičius nuo 0 iki $k = 10$.

Tam, kad perduotų tokį žymėjimą:

Indeksas	Žymė
0	6
1	2
2	9
3	3
4	7

funkcija `label` turėtų grąžinti $[6, 2, 9, 3, 7]$. Žemiau esantis paveikslėlis rodo indeksus (kairėje) bei priskirtas žymes (dešinėje).



Tarkime, kad žymės priskirtos taip, kaip nurodyta aukščiau, ir panagrinėkime tokį iškvietimą:

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

Tai reiškia, kad stotelės, turinčios paketą, žymė yra 9, o paskirties stotelės žymė yra 6. Stotelių žymės kelyje iki paskirties stotelės yra $[9, 2, 6]$. Taigi funkcija turėtų grąžinti 2, nes tai yra stotelės, kuriai turėtų būti perduotas paketas, žymė (stotelės indeksas yra 1).

Panagrinėkime kitą funkcijos iškvietimą:

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

Funkcija turėtų grąžinti 3, nes paskirties stotelė, turinti žymę 3, yra stotelės su žyme 2 kaimynė ir todėl turėtų gauti paketą tiesiogiai.

Ribojimai

- $1 \leq r \leq 10$

Kiekvienam funkcijos `label` iškvietimui:

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

Kiekvienam funkcijos `find_next_station` iškvietimui pradiniai duomenys paimami iš atsitiktinio funkcijos `label` iškvietimo:

- s ir t yra dviejų skirtingų stotelių žymės.
- c yra visų stotelės su žyme s kaimynių žymių seka, išrikiuota didėjimo tvarka

Kiekvienam testui visų funkcijai `find_next_station` perduotų masyvų c bendras dydis neviršija $100\ 000$ per visus scenarijus.

Dalinės užduotys

1. (5 taškai) $k = 1000$, nei viena stotelė neturi daugiau nei 2 kaimynių.
2. (8 taškai) $k = 1000$, jungtis i jungia stoteles $i + 1$ ir $\left\lfloor \frac{i}{2} \right\rfloor$.
3. (16 taškų) $k = 1\ 000\ 000$, daugiausiai viena stotelė turi daugiau nei 2 kaimynes.
4. (10 taškų) $n \leq 8$, $k = 10^9$
5. (61 taškas) $k = 10^9$

5-oje dalinėje užduotyje galite gauti dalinius taškus. Tegul m žymi didžiausią iš visų scenarijų funkcijos `label` gražintų žymių vertę. Jūsų gaunami taškai skaičiuojami pagal šią lentelę:

Didžiausios žymės vertė	Gaunami taškai
$m \geq 10^9$	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

Pavyzdinė vertinimo programa

Pavyzdinė vertinimo programa nuskaito įvestį šiuo formatu:

- 1 – oji eilutė: r

Toliau yra r blokų, atitinkančių kiekvieną scenarijų. Kiekvieno bloko formatas yra toks:

- 1 – oji eilutė: $n \ \ k$
- $2 + i$ – oji eilutė ($0 \le i \le n - 2$): $u[i] \ \ v[i]$
- $1 + n$ – oji eilutė: q, funkcijos `find_next_station` iškvietimų skaičius.
- $2 + n + j$ – oji eilutė ($0 \le j \le q - 1$): $z[j] \ \ y[j] \ \ w[j]$: stotelių **indeksai** j-ajam funkcijos `find_next_station` iškvietimui. Stotelė $z[j]$ turi paketą, stotelė $y[j]$ yra to paketo paskirties stotelė, o stotelė $w[j]$ yra stotelė, kuriai paketas turėtų būti perduotas.

Pavyzdinė vertinimo programa išspausdina atsakymą šiuo formatu:

- 1 – oji eilutė: m

Toliau yra r blokų, atitinkančių iš eilės einančius scenarijus, nurodytus įvestyje. Šių blokų formatas yra toks:

- $1 + j$ – oji eilutė ($0 \le j \le q - 1$): **indeksas** stotelės, kurios **žymė** grąžinta j-ojo funkcijos `find_next_station` iškvietimo šiame scenarijuje.

Atkreipkite dėmesį, kad kiekvienas pavyzdinės vertinimo programos paleidimas iškviečia tiek funkciją `label`, tiek funkciją `find_next_station`.