

Jaamad (stations)

Singapuri tuumvõrk (Singapore Internet Backbone, SIB) koosneb n jaamast **indeksitega** $0 \dots n - 1$. Nende vahel on $n - 1$ kahe-suunalist otseühendust, mis on nummerdatud $0 \dots n - 2$. Iga otseühendus ühendab omavahel kaks erinevat jaama. Kaht otseühendusega ühendatud jaama nimetatakse naabriteks.

Teekonnaks jaamast x jaama y nimetame paarikaupa erinevatest jaamadest koosnevat jada a_0, a_1, \dots, a_p , kus $a_0 = x$, $a_p = y$ ja jada iga kaks järjestikust jaama on naabrid. Igast jaamast x igasse teise jaama y on **täpselt üks** teekond.

Mistahes jaam x võib luua paketi (hulga andmeid) ja saata selle mistahes teise jaama y . Jaama y nimetatakse siis paketi **sihtjaamaks**. Paketti suunatakse mööda üheselt määratud teekonda jaamast x jaama y järgmisel viisil. Vaatleme jaama z , mille käes on hetkel pakett, mille sihtjaam on y (kus $z \neq y$). Jaam z teeb järgmist:

1. Käivitab **suunamisprotseduuri**, mis leiab z selle naabri, mis on üheselt määratud teekonnal jaamast z jaama y .
2. Suunab paketi edasi leitud naaberjaama.

Jaamade mälu kasutus on aga piiratud, seega ei saa nad meeles hoida kogu SIB struktuuri.

Sinu ülesandeks on realiseerida SIB jaoks suunamisskeem, mis koosneb kahest funktsioonist.

- Esimene funktsioon saab sisendina arvu n , SIB otseühenduste nimekirja ja täisarvu $k \geq n - 1$ ning peab määrama igale jaamale **unikaalse** täisarvulise **sildi** lõigust $0 \dots k$.
- Teine funktsioon on suunamisprotseduur, mis paigaldatakse kõikidele jaamadele pärast siltide määramist. Sisendina antakse sellele **ainult** järgmised väärtused:
 - s , praegu paketti hoidva jaama **silt**,
 - t , paketi sihtjaama **silt** (kus $t \neq s$),
 - c , jaama s naabrite **siltide** loetelu.

See funktsioon peaks tagastama selle s naabri **sildi**, kuhu pakett edasi suunata.

Ühes alamülesandes sõltub lahenduse skoor jaamadele pandud siltide maksimumist (üldiselt: mida väiksem, seda parem).

Realisatsioon

Lahendusena tuleb realiseerida järgmised funktsioonid:

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

- n on SIB jaamade arv.
- k on suurim lubatud sildi väärtus.
- u ja v on $(n - 1)$ -elemendilised massiivid, mis kirjeldavad otseühendusi. Iga otseühendus i (kus $0 \leq i \leq n - 2$) ühendab jaamad indeksitega $u[i]$ ja $v[i]$.
- Funktsioon peaks tagastama n -elemendilise massiivi L , kus $L[i]$ ($0 \leq i \leq n - 1$) on jaamale i määratud silt. Massiivi L elemendid peavad olema paarikaupa erinevad väärtused lõigust $0 \dots k$.

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

- s on selle jaama silt, mille käes pakett praegu on.
- t on paketi sihtjaama silt.
- c on massiiv, mis koosneb s kõikide naabrite siltidest. Massiiv c on sorteeritud kasvavalt.
- Funktsioon peaks tagastama s selle naabri sildi, kuhu pakett edasi saata.

Igas testis võib olla mitu erinevat omavahel sõltumatut olukorda (s.t. erinevat SIB kirjeldust). Testis, kus uuritakse r olukorda, käivitatakse eelnevaid funktsioone sisaldavat **programmi** täpselt kaks korda.

Esimesel käivitamisel:

- kutsutakse funktsiooni `label` välja täpselt r korda;
- tagastatud sildid salvestatakse hindamissüsteemi;
- funktsiooni `find_next_station` välja ei kutsuta.

Teisel käivitamisel:

- kutsutakse korduvalt välja funktsiooni `find_next_station`, kus igas kutses vastavad sisendid ühele programmi esimese käivituse **valalt valitud** olukorras tagastatud siltide skeemile;
- funktsiooni `label` välja ei kutsuta.

Muuhulgas märgime, et programmi esimese käivitamise jooksul staatilistesse või globaalsetesse muutujatesse salvestatud informatsioon ei ole `find_next_station` kutsumise ajal enam seal.

Näide

Vaatleme kutset

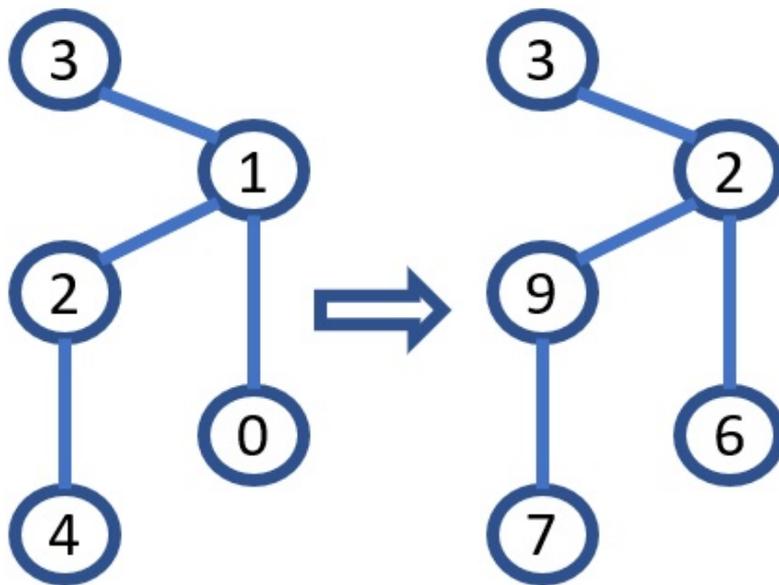
```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

Kokku on 5 jaama ja 4 otseühendust, mis ühendavad omavahel jaamade paarid indeksitega $(0, 1)$, $(1, 2)$, $(1, 3)$ ja $(2, 4)$. Sildid võivad olla täisarvud 0 kuni $k = 10$.

Siltide paigutuse

Indeks	Silt
0	6
1	2
2	9
3	3
4	7

teatamiseks peaks funktsioon `label` tagastama $[6, 2, 9, 3, 7]$. Numbrid alloleval joonisel näitavad indekseid (vasakul pool) ja silte (paremal pool).



Oletame, et jaamadele pannakse sellised sildid, nagu ülal kirjeldatud. Vaatleme kutset

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

See tähendab, et pakett on jaamas sildiga 9 ja sihtjaama silt on 6. Sihtjaama viiva teekonna jaamade sildid on $[9, 2, 6]$. Pakett tuleks suunata jaama sildiga 2 (mille indeks on 1), seega peaks funktsioon tagastama 2.

Vaatleme veel kutset

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

Et sihtjaam sildiga 3 on jaama sildiga 2 naaber ehk sihtjaam peaks paketi otse kätte saama, peaks funktsioon tagastama 3.

Piirangud

- $1 \leq r \leq 10$.

Funktsiooni `label` igal kutsel:

- $2 \leq n \leq 1\,000$.
- $k \geq n - 1$.
- $0 \leq u[i], v[i] \leq n - 1$ iga $0 \leq i \leq n - 2$ korral.

Funktsiooni `find_next_station` iga kutse sisend tuleb ühest vabalt valitud eelnevast funktsiooni `label` kutsest. Vaatleme silte, mida `label` tagastas. Siis:

- s ja t on kahe erineva jaama sildid.
- c koosneb kõikide sildiga s jaama naabrite siltidest kasvavas järjekorras.

Üheski testis ei ole funktsiooni `find_next_station` argumendiks antud massiivide pikkuste summa (kõikide olukordade peale kokku) üle 100 000.

Alamülesanded

1. (5 punkti) $k = 1\,000$ ja ühelgi jaamal ei ole üle 2 naabri.
2. (8 punkti) $k = 1\,000$ ning otseühendus i ühendab jaamad $i + 1$ ja $\lfloor \frac{i}{2} \rfloor$.
3. (16 punkti) $k = 1\,000\,000$ ja ülimalt ühel jaamal on üle 2 naabri.
4. (10 punkti) $n \leq 8$, $k = 10^9$.
5. (61 punkti) $k = 10^9$.

Alamülesandes 5 on osaline hindamine. Olgu m maksimaalne silt, mida `label` üle kõikide olukordade produtseerib. Selle alamülesande eest antav skoor arvutatakse alloleva tabeli alusel:

Suurim silt	Skoor
$m \geq 10^9$	0
$2\,000 \leq m < 10^9$	$50 \cdot \log_{5 \cdot 10^5} \left(\frac{10^9}{m} \right)$
$1\,000 < m < 2\,000$	50
$m \leq 1\,000$	61

Näidishindaja

Näidishindaja loeb sisendit järgmises vormingus:

- Rida 1: r .

Järgnevad r plokki, millest igaüks kirjeldab ühte olukorda. Plokid on järgmises vormingus:

- Rida 1: $n \ k$.
- Rida $2 + i$ (kus $0 \leq i \leq n - 2$): $u[i] \ v[i]$.
- Rida $1 + n$: q , funktsiooni `find_next_station` kutsete arv
- Rida $2 + n + j$ (kus $0 \leq j \leq q - 1$): $z[j] \ y[j] \ w[j]$, funktsiooni `find_next_station` j . kutse jaamade **indeksid**, mis näitavad, et pakett asub jaamas $z[j]$, paketi sihtjaam on $y[j]$ ja pakett tuleks suunata jaama $w[j]$.

Näidishindaja trükib väljundi järgmises vormingus:

- Rida 1: m .

Sellele järgneb r plokki, mis vastavad sisendis antud olukordadele. Plokid on järgmises vormingus:

- Rida $1 + j$ (kus $0 \leq j \leq q - 1$): selle jaama **indeks**, mille **sildi** tagastas funktsiooni `find_next_station` antud olukorra j . kutse.

Märgime, et näidishindaja iga käivitus kutsub välja nii funktsiooni `label` kui funktsiooni `find_next_station`.