

## O'yin

$n$  ta sayyorani aniqlab bo'lgach (sayyoralar 0 dan  $n - 1$  gacha raqamlangan) fir'avnlr ular orasida **bir tomonlamali** teleporterlar yordamida teleportatsiya sistemasini qurishni boshladilar. Har bir teleporterning boshlanish va tugash sayyorasi mavjud. Turist teleporterni boshlanish sayyorasida ishlatganida, u tugash sayyorasiga teleportatsiya qilinadi. Eslatma: teleporterning boshlanish ham tugash sayyoralari bir xil bo'lishi mumkin. Boshlanish sayyorasi  $u$  va tugash sayyorasi  $v$  bo'lgan teleporter  $(u, v)$  orqali ifodalanadi.

Teleportatsiya tizimini yoyilishini qo'llab quvvatlash uchun fir'avnlr teleporterlar orqali sayohat paytida o'ynash mumkin bo'lgan o'yinni yaratdilar. Sayohatchi o'yinni istalgan sayyorada boshlashi mumkin.  $0, 1, \dots, k - 1$  ( $k \leq n$ ) sayyoralar **mahsus sayyoralar** sanaladi. Sayoharchi har safar shu sayyoralariga kirganda u marka oladi.

Ayni paytda, har bir  $i$  ( $0 \leq i \leq k - 2$ ) lar uchun  $(i, i + 1)$  teleporteri mavjud. Bu  $k - 1$  ta teleporterlar **boshlang'ich teleporterlar** deb ataladi.

Yangi teleporterlar bittalab qo'shiladi. Yangi yo'llar qo'shilgani sari sayohatchida cheksiz markalar olish imkoni paydo bo'lishi mumkin. Aniqroq aytganda, bu  $w[0], w[1], \dots, w[t]$  sayyorlari quyidagi shartlarni bajarganda sodir bo'ladi:

- $1 \leq t$
- $0 \leq w[0] \leq k - 1$
- $w[t] = w[0]$
- Har bir  $i$  ( $0 \leq i \leq t - 1$ ) uchun,  $(w[i], w[i + 1])$  teleporteri mavjud.

Eslatma: sayohatchi **istalgan** boshlang'ich teleporterni, shuningdek hozirgacha qo'shilgan istalgan teleporterdan foydalanishi mumkin.

Sizning vazifangiz fir'avnga har safar yangi yo'l qo'shilganda sayohatchi cheksik markalar olishi mumkin yoki yo'qligini tasdiqlashda yordam berishdir.

## Bajarish tafsilotlari

Siz quyidagi protsedurani amalga oshirishingiz kerak:

```
init(int n, int k)
```

- $n$ : sayyorlar soni.

- $k$ : mahsus sayyorlar soni.
- Bu protsedura `add_teleporter` so'rovlaridan avval faqat bir martagina chaqiriladi.

```
int add_teleporter(int u, int v)
```

- $u$  va  $v$ : qo'shilgan teleporterning boshlanish va tugash sayyorolari.
- Bu funksiya maksimal  $m$  marta chaqiriladi ( $m$  ning qiymatlari uchun Chegaralar qismini ko'ring).
- Agarda shu teleporter qo'shilgandan keyin sayohatchi cheksiz markalar olishi mumkin bo'lsa bu funksiya 1, ask holda 0 qaytarishi lozim.
- Funksiya 1 qaytargach dastur yakunlanadi.

## Misollar

### Misol

Quyidagi so'rovga nazar soling:

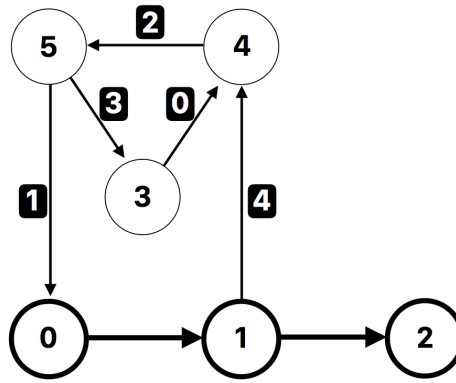
```
init(6, 3)
```

Bu misolda, jami 6 ta sayyora va 3 ta maxsus sayyora mavjud. 0, 1, va 2 sayyoralar mahsus sayyoralaridir. Boshlang'ich teleporterlar (0, 1) va (1, 2) lardir.

Faraz qiling grader quyidagi so'rovlarni chaqiradi:

- (0) `add_teleporter(3, 4)`: Siz 0 qaytarishingiz lozim.
- (1) `add_teleporter(5, 0)`: Siz 0 qaytarishingiz lozim.
- (2) `add_teleporter(4, 5)`: Siz 0 qaytarishingiz lozim.
- (3) `add_teleporter(5, 3)`: Siz 0 qaytarishingiz lozim.
- (4) `add_teleporter(1, 4)`: Bu holatda cheksiz markalar to'plash mumkin. Masalan, sayohatchi 0 - sayyorada o'z sayohatini boshlaydi, va 1, 4, 5, 0, 1, 4, 5, 0, ... sayyoralariga huddi shu tartibda boradi. Shu sababdan ham siz 1 qaytarishingiz lozim, va sizning dasturingiz yakunlanadi.

Quyidagi diagramma bu misolni tasvirlaydi. Mahsus sayyoralar va boshlang'ich teleporterlar qoraytirib ko'rsatilgan. `add_teleporter` orqali qo'shilgan teleporterlar 0 dan 4 gacha shu tartibda belgilangan.



## Misol 2

Quyidagi so'rovga nazar soling:

```
init(4, 2)
```

Bu misolda, jami 4 ta sayyora va 2 ta maxsus sayyora mavjud. 0 va 1 sayyoralar mahsus sayyoralardir. Boshlang'ich teleporter (0, 1) dir.

Faraz qiling grader quyidagi so'rovni chaqiradi:

- `add_teleporter(1, 1)`: (1,1) teleporteri qo'shilgach, cheksiz markalar to'plash imkoni mavjud. Masalan, sayohatchi 1 - sayyorada o'z sayohatini boshlaydi, va 1 sayyorasiga cheksiz marotaba (1,1) teleporterini ishlatgan holda kiradi. Shu sababdan ham siz 1 qaytarishingiz lozim, va sizning dasturingiz yakunlanadi.

Yana boshqa kiritish/chiqarish misoli attachment to'plamida mavjud.

## Chegaralar

- $1 \leq n \leq 300\,000$
- $1 \leq m \leq 500\,000$
- $1 \leq k \leq n$

`add_teleporter` protsedurasidagi har bir so'rov uchun:

- $0 \leq u \leq n - 1$  va  $0 \leq v \leq n - 1$
- $(u, v)$  teleporterini qo'shishdan avval  $u$  sayyorasidan  $v$  sayyorasiga teleporter mavjud emas.

## Qism-masalalar

1. (2 points)  $n = k$ ,  $n \leq 100$ ,  $m \leq 300$
2. (10 points)  $n \leq 100$ ,  $m \leq 300$
3. (18 points)  $n \leq 1\,000$ ,  $m \leq 5\,000$
4. (30 points)  $n \leq 30\,000$ ,  $m \leq 50\,000$ ,  $k \leq 1\,000$

5. (40 points) Qo'shimcha shartlar yo'q

## Sample grader

Sample grader kiruvchi ma'lumotlarni quyidagicha qabul qiladi:

- 1 - qator:  $n\ m\ k$
- $2 + i$  ( $0 \leq i \leq m - 1$ ) qatorlar:  $u[i]\ v[i]$

Sample grader avval `init` funksiyasini chaqiradi, so'ngra  $i = 0, 1, \dots, m - 1$  ketma-ketlikda  $u = u[i]$  va  $v = v[i]$  `add_teleporter` funksiyasini

Sample grader `add_teleporter` so'rovlari orasida birinchi bo'lib 1 qaytargan so'rov indeksini qaytaradi (which is between 0 and  $m - 1$ , inclusive), yoki, agarda hamma so'rovlar 0 qaytargan bo'lsa  $m$  qaytaradi.

Agarda qaysidir `add_teleporter` so'rovi 0 yoki 1 dan boshqa son qaytarsa, sample grader -1 chiqaradi va sizning dasturingiz zudlik bilan yakunlanadi.