

Тоглоом

0-ээс $n - 1$ гэж дугаарлагдсан n ширхэг гаригуудыг Фараон нээн илрүүлснийхээ дараа тэдгээрийн хооронд **нэг-замын телепортлогчууд**-ыг тээврийн систем болгон байгуулж эхлэв. Телепортлогч бүр эхлэлийн гариг, төгсгөлийн гаригтай. Хэрэв жуулчин телепортлогчийг эхлэлийн гаригаас хэрэглэвэл тэрээр төгсгөлийн гаригт хүрнэ. Эхлэлийн гариг болон төгсгөлийн гаригууд адил байж болохыг анхаараарай. Эхлэлийн гариг u , төгсгөлийн гариг v байх телепортлогчийг (u, v) гэж дүрсэлнэ.

Телепортын системийн өргөн хэрэглээг дэмжих үүднээс жуулчид уг тээврийн системээр аялж байх үедээ тоглож болохоор тоглоомыг Фараон бодож олов. Жуулчин аль ч гаригаас эхэлж болно. Гариг $0, 1, \dots, k - 1$ ($k \leq n$) нь **онцгой гариг** болно. Жуулчин эдгээр онцгой гаригт орох бүрдээ тэрээр тамгалуулах ёстой.

Одоогоор i ($0 \leq i \leq k - 2$) гараг бүрд $(i, i + 1)$ гэсэн телепортлогч байгаа. Эдгээр $k - 1$ телепортлогчуудыг **эхлэлийн телепортлогчид** гэж нэрлэнэ.

Шинэ телепортлогчууд нэг нэгээр нэмэгдэнэ. Шинэ телепортлогчууд нэмэгдэх үед жуулчин хязгааргүй удаа тамгалуулах боломжтой болж болно. Тодорхойлбол дараах нөхцөлийг хангах $w[0], w[1], \dots, w[t]$ гэсэн гаригуудын дараалал байх үед ийм зүйл тохиолдох боломжтой:

- $1 \leq t$
- $0 \leq w[0] \leq k - 1$
- $w[t] = w[0]$
- Бүх i ($0 \leq i \leq t - 1$) бүрийн хувьд $(w[i], w[i + 1])$ гэсэн телепортлогч оршин байх үед.

Жуулчин эхлэлийн телепортлогчдыг болон аль хэдийн нэмэгдэн **аль ч** телепортлогчийг ашиглах боломжтойг анхаар.

Таны даалгавар бол шинэ телепортлогч бүрийг нэмсний дараа жуулчин хязгааргүй их тамга цуглуулах боломжтой эсэхийг шалгахад Фараонд туслах юм.

Хэрэгжүүлэх заавар

Та дараах процедуруудыг хэрэгжүүлэх ёстой:

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

- n : Гаригуудын тоо.

- k : Тусгай гаригуудын тоо.
- Энэ процедур ямар ч `add_teleporter` процедур дуудагдаагүй байхад ганц л удаа дуудагдана.

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

- u, v : Нэмэгдэж буюу телепортлогчийн эхлэлийн ба төгсгөлийн гаригууд.
- Уг функц хамгийн ихдээ m удаа (m -ын хэмжээг мэдэхийн тулд хязгаарлалт хэсгийг хар) дуудагдана.
- Хэрэв (u, v) телепортлогчийг нэмсний дараа жуулчин хязгааргүй их тамга цуглуулах боломжтой бол уг функц 1-ийг буцаана. Үгүй бол уг функц 0-ийг буцаана.
- Хэрэв уг функц 1-ийг буцаавал таны программыг зогсоох болно.

Жишээ

Жишээ 1

Дараах функц дуудалтын хувьд:

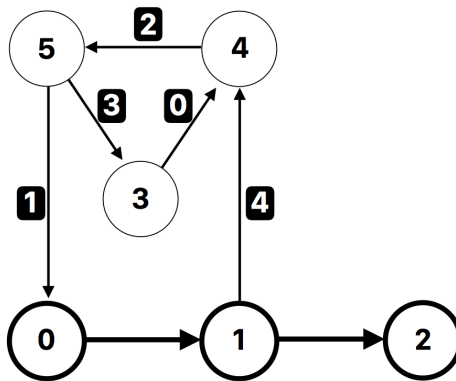
```
init(6, 3)
```

Уг жишээнд 6 гариг байх ба эдгээрээс 3 нь тусгай гаригууд. Гариг 0, 1, 2 нь тусгай гаригууд. Эхлэлийн телепортлогчид нь $(0, 1)$ болон $(1, 2)$.

Шалгагч дараах дуудалтуудыг хийсэн гэж үзье:

- (0) `add_teleporter(3, 4)`: Та 0-ийг буцаах ёстой.
- (1) `add_teleporter(5, 0)`: Та 0-ийг буцаах ёстой.
- (2) `add_teleporter(4, 5)`: Та 0-ийг буцаах ёстой.
- (3) `add_teleporter(5, 3)`: Та 0-ийг буцаах ёстой.
- (4) `add_teleporter(1, 4)`: Энэ үед хязгааргүй их тамга цуглуулах боломжтой болно. Жишээ нь 0 гаригаас эхэлсэн жуулчин 1, 4, 5, 0, 1, 4, 5, 0, ... гэсэн дарааллаар гаригуудад очих боломжтой. Тиймээс таны та 1-ийг буцааж таны программыг зогсоох болно.

Дараах зурагт уг жишээг харуулав. Тусгай болон эхлэлийн гарагуудыг тодоор дүрсэлсэн. `add_teleporter`-оор нэмэгдэн телепорлогчуудыг дарааллаар нь 0 through 4 гэж дугаарласан.



Жишээ 2

Дараах функц дуудалтын хувьд:

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

Уг жишээнд 4 гариг байх ба эдгээрээс 2 нь тусгай гаригууд. Гариг 0, 1 нь тусгай гаригууд. Эхлэлийн телепортлогч нь (0, 1).

Шалгагч дараах дуудалтуудыг хийсэн гэж үзье:

- `add_teleporter(1, 1)`: (1,1) телепортлогчийг нэмсний дараагаар хязгааргүй их тамга цуглуулах боломжтой болно. Жишээ нь жуулчин 1 гарагаас эхлэн 1 гарагт хязгааргүй олон удаа (1,1) телепортлогчийг ашиглан хүрэх боломжтой болно. Тиймээс таны та 1-ийг буцааж таны программыг зогсоох болно.

Өөр нэг жишээ оролт/гаралт хавсаргасан илгээмжид байгаа.

Хязгаарлалт

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

Порецедур `add_teleporter`-ийн дуудалт бүрийн хувьд:

- $0 \leq u \leq n - 1$ болон $0 \leq v \leq n - 1$
- (u, v) телепортлогчийг нэмэхээс өмнө u гаригаас v гарагт хүрэх телепортлогч байхгүй.

Дэд даалгавар

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) Ямар ч нэмэлт хязгаарлалт байхгүй.

Өгөгдсөн шалгагч программ

Өгөгдсөн шалгагч программ нь оролтыг дараах форматаар уншдаг:

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

Өгөгдсөн шалгагч нь эхлээд `init`, дараагаар нь `add_teleporter` функцийг $i = 0, 1, \dots, m - 1$ гэсэн дарааллаар $u = u[i], v = v[i]$ бүрийн хувьд дуудна.

Уг программ `add_teleporter` функц хамгийн анхны удаа 1 буцаасан индексийг (энэ нь $[0, m - 1]$ гэсэн завсарт байна) эсвэл бүх `add_teleporter` дуудалт 0 буцаасан үед m -ийг хэвлэнэ.

Хэрэв `add_teleporter` дуудсаны дараа 0, 1-ээс өөр утга буцаавал өгөгдсөн шалгагч -1 хэвлэж программыг тэр дор нь дуусгана.