

## Тоглоомын зохиомж

| Бодлогын нэр          | ToyDesign          |
|-----------------------|--------------------|
| Оролтын файл          | Интерактив бодлого |
| Гаралтын файл         | Интерактив бодлого |
| Хугацааны хязгаарлалт | 1 секунд           |
| Хугацааны хязгаарлалт | 256 мегабайт       |

Та тоглоомын зохиомж гаргадаг компанид ажилладаг. Бүтээгдэж байгаа шинэ тоглоом нь дараах байдлаар ажилладаг: 1-ээс  $n$  хүртлэх тоогоор дугаарлагдсан  $n$  ширхэг хөл хайрцагнаас цухуйж байдаг. Зарим хос хөл хоорондоо хайрцган дотор дамжуулагч утсаар холбогдсон байдаг. (Өөрөөр хэлбэл хөл болон дамжуулагч утаснууд чиглэлгүй граф үүсгэнэ. Энд хөлнүүд нь орой, утаснууд нь ирмэг болно.) Утаснууд гаднаасаа харагдахгүй ба тэдний тухай мэдээлэл олж авах ганц арга нь хөлнүүд дээр **тестер** ашиглах явдал юм: Бид  $i \neq j$  байх  $i$  ба  $j$  гэсэн хоёр хөлөн дээр тестер хэрэглэн тэднийг хайрцган дотор шууд эсвэл шууд бусаар холбогдсон эсэхийг мэдэж болно. (Ингэснээр, тестер бидэнд эдгээр хоёр хөлний хооронд графын зам байгаа эсэхийг хэлнэ.)

Хайрцган доторх холболтуудын олонлогийг тоглоомын **зохиомж** гэж нэрлэнэ.

Та эдгээр зохиомжид хүсэлт гаргаж, үүсгэх тусгай програм хангамжийг ашиглаж байгаа. Энэ програм хангамж нь дараах байдлаар ажилладаг: Эхлээд "зохиомж 0" гэж тэмдэглэх тоглоомын зохиомжоос эхэлнэ. Энэ зохиомж нь хайрцагны доторх холболтыг харуулдаггүй. Харин үүний оронд та дараах үйлдлийг дахин дахин хийж болно:

- Та  $a$ -р зохиомж болон  $i \neq j$  байх  $i$  ба  $j$  хөлний хоёр дугаарыг сонгон авна.
- Уг програм хангамж эдгээр хоёр хөл дээр тестерийг хэрэглэхэд юу болохыг хэлж өгнө. Өөрөөр хэлбэл, энэ нь  $a$ -р зохиомж дээр  $i$  болон  $j$  хөлнүүд (шууд эсвэл шууд бусаар) холбогдсон эсэхийг хэлж өгнө.
- Мөн, хэрэв хөлнүүд  $a$  зохиомж дотор шууд эсвэл шууд бусаар холбогдоогүй бол  $a$  зохиомжид байгаа бүх холболтууд дээр нэмээд  $i$  болон  $j$ -г шууд холбосон нэг нэмэлт холболт бүхий шинэ зохиомжийг үүсгэнэ. Энэ зохиомжид дараагийн боломжит дугаарыг өгнө. (Эхний зохиомжийг 1, дараагийнхийг 2 гэх мэтээр дугаарлана.) Энэ нь  $a$

загварыг өөрчлөхгүй бөгөөд зөвхөн шинэ холболт бүхий шинэ зохиомжийг үүсгэнэ гэдгийг анхаараарай.

Таны зорилго бол 0-р зохиомжийн талаар энэ үйлдлийг ашиглан аль болох ихийг мэдэх явдал юм.

Заримдаа зохиомж 0-ийн холболтуудын олонлогийг бүрэн тодорхойлох боломжгүй байдаг гэдгийг анхаараарай. Учир нь шууд болон шууд бус холболтыг ялгах аргагүй юм. Жишээ нь,  $n = 3$  байх үеийн хоёр зохиомжийг авч үзье:



Тестер хоёр зохиомжийн аль алиных нь хувьд дурын хоёр хөлийг холбоотой гэж гаргана. Иймд бид энэ хоёр зохиомжийг уг програмыг ашиглан ялгаж чадахгүй.

Таны зорилго бол зохиомж 0-той эквивалент байх ямар нэг зохиомжийг олох явдал юм. Хэрэв хоёр зохиомжийн хувьд тестер бүх хос хөлнүүд дээр ижил үр дүн гаргаж байвал эдгээрийг **эквивалент** зохиомж гэнэ.

## Хэрэгжүүлэлт

Энэ бол интерактив бодлого. Та дараах функцийг хэрэгжүүлнэ.

```
void ToyDesign(int n, int max_ops);
```

функц нь зохиомж 0-той эквивалент байх зохиомжийг тодорхойлно. Таны хэрэгжүүлэлт хоёр үйлдлийг тэг эсвэл олон удаа гүйцэтгэх замаар зорилгодоо хүрнэ. Эхний үйлдлээр таны програм дараах функцийг дуудна:

```
int Connected(int a, int i, int j);
```

энд  $1 \leq i, j \leq n, i \neq j, a \geq 0$  ба  $a$  нь одоог хүртэл зохиогдсон загварын тооноос хэтрэх ёсгүй. Хэрэв  $i$  болон  $j$  хөлнүүд нь (шууд болон шууд бусаар) зохиомж  $a$  дотор холбогдсон тохиолдолд бид  $a$ -г буцаана. Эсрэг тохиолдолд, энэ нь одоог хүртэл зохиогдсон зохиомжуудын тоон дээр нэгийг нэмж буцаана. Энэ нь шинээр үүсгэж байгаа зохиомжийн дугаар болох ба уг зохиомжийн дотор зохиомж  $a$ -гийн бүх холболтоос гадна  $i$  ба  $j$ -гийн хоорондох холбоос байна. `Connected` функцийг дээд тал нь `max_ops` дуудаж болно.

Таны програм `Connected` функцээ ашиглах тоогоо дуусгасан бол зохиомж 0-той эквивалент зохиомжийг тодорхойлж, ажиллагаагаа хэвийн дуусгах ёстой. Зохиомжийг тодорхойлохын тулд уг програм дараах дуудалтыг хийнэ:

```
void DescribeDesign(std::vector<std::pair<int,int>> result);
```

`result` параметр нь хөл хоорондын шууд холболтыг илэрхийлэх хос бүхэл тоонуудас тогтох вектор байна. Хос бүр нэг холболтонд харгалзах ба хоёр ширхэг хөлний дугаар агуулна. Хос (эрэмбэлэгдээгүй) хөл бүрийн хувьд дээд тал нь нэг холболт байх ба ямар ч хөл өөртэйгөө шууд холбогдохгүй.

## Хязгаарлалт

- $2 \leq n \leq 200$

## Оноо

- Subtask 1 (10 points):  $n \leq 200$ ,  $max\_ops = 20\,000$
- Subtask 2 (20 points):  $n \leq 8$ ,  $max\_ops = 20$
- Subtask 3 (35 points):  $n \leq 200$ ,  $max\_ops = 2\,000$
- Subtask 4 (35 points):  $n \leq 200$ ,  $max\_ops = 1\,350$

## Жишээ харилцаа

| Оролцогчийн үйлдэл              | Шалгагчийн үйлдэл            | Тайлбар  |
|---------------------------------|------------------------------|--|
|                                 | <code>ToyDesign(4, 5)</code> | Тоглоом 4 хөлтэй. Дээд тал нь 5 удаа <code>Connected</code> функцийг дуудаж зохиомж 0-той эквивалент зохиомжийг тодорхойлно. |
| <code>Connect(0, 1, 2)</code>   | Returns 1.                   | Зохиомж 0 дээр Хөл 1 ба 2 нь шууд болон шууд бусаар холбогдоогүй. Шинэ зохиомж 1-ийг үүсгэнэ.                                |
| <code>Connected(1, 3, 2)</code> | Returns 2.                   | Хөл 3 ба 2 нь зохиомж 1 дээр шууд болон шууд бусаар холбогдоогүй. Шинэ зохиомж 2-ыг үүсгэнэ.                                 |
| <code>Connected(0, 3, 4)</code> | Returns 0.                   | Хөл 3 ба 4 нь зохиомж 0 дээр шууд болон шууд бусаар холбогдсон. Шинэ зохиомж үүсгэхгүй.                                      |

```
DescribeDesign({{3, 4}})
```

-

Бид ганц холболттой загварыг тодорхойлно: Хөл 3 ба 4.

## Жишээ шалгагч

Бодлогын хавсралтанд байгаа `ToyDesign.zip` файл доторх `grader.cpp` шалгагч нь оролтоо дараах хэлбэрээр уншина:

- Эхний мөрөнд зангилаануудын тоо болох  $n$ , ирмэгүүдийн тоо  $m$  болон  $max\_ops$  өгөгдөнө.
- Дараагийн  $m$  мөрөнд ирмэгүүдийг зангилааны хос байдлаар өгнө.

Жишээ шалгагч оролтыг уншаад `ToyDesign` функцийг дуудна.

Жишээ шалгагчийг таны бодолттой хөрвүүлэхийн тулд терминал дээр доорх командыг ашигла:

```
g++ -std=gnu++11 -O2 -o solution grader.cpp solution.cpp
```

энд `solution.cpp` нь таны CMS руу явуулах код. Програмаа жишээ оролтон дээр ажиллуулахын тулд терминал дээр дараах командыг ашиглана:

```
./solution < input.txt
```