

Oyuncaq Dizaynı

Məsələnin Adı	Oyuncaq Dizaynı
Giriş Faylı	İnteraktiv Tapşırıq
Çıxış Faylı	İnteraktiv Tapşırıq
Zaman limiti	1 saniyə
Yaddaş limiti	256 MB

Oyuncaq istehsal edən bir şirkətdə işləyirsiniz. Yeni yaradılan oyuncaq belə işləyir: Qutudan çıxan n sayda sancaq var. Bu sancaqlar 1-dən n -ə tam ədədlərlə nömrələnib. Bəzi cüt sancaqlar qutunun içində naqillərlə birləşdirilib. Başqa sözlə desək, sancaqlar və naqillər istiqamətlənməmiş (undirected) qraf əmələ gətirir. Burada sancaqlar təpələri (vertices), naqillər isə əlaqələri (edges) bildirir. Naqillər kənarından görünür və onlar haqqında nəşə öyrənməyin yeganə yolu sancaqlar üzərində **tester**dən istifadə etməkdir: Biz iki sancaq i və j ($i \neq j$) seçə bilirik və tester bu iki sancağın qutunun içərisində birbaşa və ya dolay yolla naqillərlə əlaqəli olub olmadığını deyəcək. (Beləliklə, tester qrafda həmin təpələr arasında yolun olub olmadığını bildirir.)

Qutunun içindəki birləşmələr çoxluğunu oyuncağın **dizaynı** adlandıracağıq.

Bu dizaynları yaratmaq və onlara sorğu göndərmək üçün xüsusi proqram təminatından istifadə edirsiniz. Bu proqram təminatı belə işləyir: Oyuncağın, bizim "dizayn 0" kimi adlandırdığımız bir dizaynından başlayır. O, bu dizayn üçün qutunun içindəki naqilləri göstərmir. Bunun əvəzinə aşağıdakı üç mərhələli əməliyyatı təkrar-təkrar yerinə yetirə bilərsiniz:

1. Siz dizayn nömrəsini, a və iki sancaq nömrəsini, i və j ($i \neq j$) seçirsiniz.
2. Proqram təminatı sizə bu iki sancaq üzərində tester istifadə etsək nə baş verəcəyini deyir. Başqa sözlə, i və j sancaqlarının (birbaşa və ya dolay yolla) a dizaynında naqillərlə əlaqəli olub olmadığını sizə bildirir.
3. Həmçinin, əgər sancaqlar a dizaynında birbaşa və ya dolay yolla bağlı deyildisə, o zaman a dizaynından bütün əlaqələrə üstəgəl i və j arasında birbaşa əlaqəyə malik olan yeni dizayn yaranır. Bu dizayna növbəti mövcud dizayn nömrəsi verilir. (Beləliklə, bu şəkildə yaradılan ilk dizayna 1, sonrakına 2 və s. nömrələr veriləcək.) Qeyd edək ki, bu a dizaynını dəyişmir, sadəcə əlavə əlaqəyə malik yeni dizayn yaradır.

Məqsədiniz bu əməliyyatdan istifadə etməklə dizayn 0 haqqında mümkün qədər çox öyrənməkdir.

Nəzərə alın ki, dizayn 0-dakı əlaqələri dəqiqliklə müəyyən etmək həmişə mümkün olmaya bilər. Çünki, birbaşa və dolayı yolla əlaqəni ayırd etmək mümkün deyil. Məsələn, $n = 3$ olan aşağıdakı iki dizayna nəzər yetirin:



Tester hər iki dizaynda istənilən iki sancağın əlaqəli olduğunu deyəcək. Beləliklə, yuxarıda təsvir olunan proqram təminatından istifadə etməklə onları fərqləndirə bilmərik.

Məqsədiniz dizayn 0-a ekvivalent olan hər hansı dizaynı müəyyən etməkdir. Tester bütün sancaq cütləri üçün hər iki dizaynda eyni nəticəni bildirirsə, bu iki dizayn **ekvivalent**dir.

İmplementasiya detalları

Bu interaktiv məsələdir. Aşağıdakı funksiyanı implementasiya etməlisiniz:

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

Bu funksiya dizayn 0-a *ekvivalent* olan bir dizayn müəyyən edir. Sizin implementasiyanız aşağıda təsvir olunan iki funksiyanı çağırmaqla bunu əldə etməlidir. Çağıra biləcəyiniz birinci funksiya:

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

Burada $1 \leq i, j \leq n, i \neq j, a \geq 0$, və a indiyə kimi yaradılmış dizaynların sayını aşmamalıdır. Əgər dizayn a -da i və j sancaqları (birbaşa və ya dolayı yolla) bağlı olarsa, o zaman funksiya a qaytaracaq. Əks halda o, indiyə kimi olan dizaynların sayı üstəgəl bir qaytaracaq və bu ədəd a dizaynından bütün əlaqələrə üstəgəl i və j arasında birbaşa əlaqəyə malik olan yeni dizaynın nömrəsi olacaq. `Connected` funksiyası ən çox `max_ops` dəfə çağırıla bilər.

Proqramınız `Connected` əməliyyatları ilə işini bitirdikdən sonra, dizayn 0-a ekvivalent olan bir dizaynı təsvir etməlidir. Dizaynı təsvir etmək üçün proqramınız aşağıdakı funksiyanı çağırmalıdır:

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

`result` parametri sancaqlar arasındakı birbaşa əlaqələri təsvir edən tam ədəd cütlərindən ibarət "vector"-dur. Hər bir cüt bir əlaqəni bildirir və əlaqənin iki sancaq nömrəsindən ibarət olmalıdır. İstənilən iki sancaq cütü (nizamsız) arasında ən çox bir birbaşa əlaqə olmalıdır və hər hansı

sancağın özündən özünə əlaqə olmamalıdır. Bu funksiyanın çağırılması proqramınızın icrasını dayandırır.

Məhdudiyyətlər

- $2 \leq n \leq 200$

Qiymətləndirmə

- Alt tapşırıq 1 (10 bal): $n \leq 200$, $max_ops = 20\,000$
- Alt tapşırıq 2 (20 bal): $n \leq 8$, $max_ops = 20$
- Alt tapşırıq 3 (35 bal): $n \leq 200$, $max_ops = 2\,000$
- Alt tapşırıq 4 (35 bal): $n \leq 200$, $max_ops = 1\,350$

İnteraksiya nümunəsi

Sizin hərəkətiniz	Qreyderin hərəkəti	İzah
	<code>ToyDesign(4, 20)</code>	Oyuncağın 4 sancağı var. Siz <code>Connected</code> funksiyasını ən çox 20 dəfə çağırmaqla dizayn 0-a ekvivalent olan hər hansı bir dizaynı müəyyən etməlisiniz.
<code>Connected(0, 1, 2)</code>	1 qaytarır.	1 və 2 sancaqları dizayn 0-da birbaşa və ya dolay yolla əlaqəli deyil. Yeni dizayn 1 yaradıldı.
<code>Connected(1, 3, 2)</code>	2 qaytarır.	3 və 2 sancaqları dizayn 1-də birbaşa və ya dolay yolla əlaqəli deyil. Yeni dizayn 2 yaradıldı.
<code>Connected(0, 3, 4)</code>	0 qaytarır.	3 və 4 sancaqları dizayn 0-da birbaşa və ya dolay yolla əlaqəlidir. Yeni dizayn yaradılmır.
<code>DescribeDesign({{3, 4}})</code>	-	Yalnız bir əlaqəsi olan bir dizaynı təsvir edirik: 3 və 4-cü sancaqlar.

Nümunə Qreyder

ToyDesign.zip əlavəsində verilmiş grader.cpp adlı nümunə qreyder giriş verilənlərini standart girişdən aşağıdakı formatda oxuyur:

- İlk sətirdə sancaqların sayı, n , birbaşa əlaqələrin sayı, m və max_ops^* verilir.
- Növbəti m sətirin hər birində birbaşa əlaqəni göstərən sancaqlar cütü verilir.

Nümunə qreyder giriş verilənlərini oxuduqdan sonra həllinizdəki ToyDesign funksiyasını çağırır. Həllinizdən asılı olaraq qreyder aşağıdakı mesajlardan birini çıxışa verəcək:

- "Wrong answer: Number of operations exceeds the limit.", Əgər Connected funksiyasına çağırışların sayı max_ops -u keçərsə.
- "Wrong answer: Wrong design id.", Əgər Connected funksiyasına göndərilən a parametri funksiyaya çağırış edilən anda hələ mövcud olmayan bir dizaynın nömrəsidirsə.
- "Wrong answer: Incorrect design.", Əgər DescribeDesign ilə təsvir edilən dizayn, dizayn 0-a ekvivalent deyilsə.
- "OK!", Əgər DescribeDesign ilə təsvir edilən dizayn, dizayn 0-a ekvivalent olarsa.

Nümunə qreyderi həllinizlə birlikdə kompayl etmək üçün terminalda aşağıdakı əmrədən istifadə edə bilərsiniz:

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

Burada solution.cpp CMS-ə göndərəcəyiniz həll faylıdır. Proqramınızı əlavədə verilmiş nümunə giriş faylı ilə işlətmək üçün terminalda aşağıdakı əmri yazın:

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