Permutation Game

Alice ve Bob beraber entellektüel oyunlar oynamayı seven çocukluk arkadaşlarıdır. Bugün graflar üzerinde yeni bir oyun oynayacaklardır.

Oyunda m köşeli (node) **connected** bir graf vardır, köşeler $0'dan\ m-1'e$ numaralandırılmıştır, ve e tane kenar, $0'dan\ e-1'e$ numaralandırılmıştır. i-inci kenar $u[i]$ ve $v[i]$ köşelerini bağlar.

Oyunda ayrıca n elemanlı bir permutasyon $p[0], p[1], \ldots, p[n-1]$ vardır, ve $m \leq n$ şartı sağlanır. Permutasyon $0'dan\ n-1'e$ her elemanın tam olarak bir kere herhangi bir sırada geçtiği bir dizidir. Permutasyon $p'nin$ **score**'u $p[i] = i$ şartını sağlayan i indislerinin sayısıdır.

Oyun en fazla 10^{100} tur sürecektir. Her turda şu olacaktır:

1. Eğer Alice oyunu bitirmeye karar verirse, oyun biter.
2. Öbür durumda, Alice $0 \leq t[i] < n$ şartını sağlayacak şekilde **farklı indisler** $t[0], t[1], \ldots, t[m-1]$ seçer. Unutmayın ki $t[0] < t[1] < \ldots < t[m-1]$ şartı **sağlanmak zorunda değildir**.
3. Bob grafın kenarlarından bir indis seçer $0 \leq j < e$ ve $p[t[u[j]]]$ ve $p[t[v[j]]]$ değerlerini değiştirir (swap).

Alice final skorunu maksimum yapmak isterken, Bob minimum yapmak ister.

Alice'e Bob'a karşı oynarken yardım edin, hamleler grader tarafından simüle edilmektedir.

optimal score'u Alice ve Bob optimal oynadığındaki son skor olarak tanımlayalım.

Permutasyonun optimal skorunu hesaplamalısınız ve daha sonra da oyunu Bob ile oynayıp birkaç tur sonunda **en azından** optimal skora ulaşmalısınız.

Unutmayın ki Alice'in stratejisi Bob ne yaparsa yapsın, optimal olmayan hamleler yaptığında bile çalışacaktır.

Kodlama Detayları

Aşağıdaki prosedürü kodlamalısınız:

```
int Alice(int m, int e, std::vector<int> u, std::vector<int> v,
          int n, std::vector<int> p)
```

- m: grafdaki köşe sayısı.
- e: grafdaki kenar sayısı.
- u ve v: e uzunluğundaki kenarları temsil eden dizi.
- n: permutasyonun uzunluğu.
- p: n elemanlı permutasyonu gösteren dizi.
- Bu prosedür kesinlikle bir kere çağırılacaktır.
- Bu prosedür sadece bir sayı dönecektir– Oyunun optimal skoru.

Bu prosedürde aşağıdaki prosedürü çağrabilirsiniz:

```
int Bob(std::vector<int> t)
```

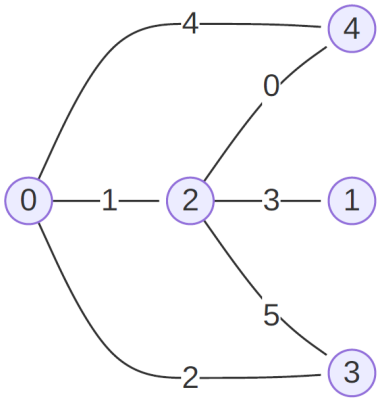
- t: m farklı eleman içeren bir dizi, $0 \leq t[i] < n$ ve her $i \neq j$ için $t[i] \neq t[j]$.
- Bu fonksiyon sadece bir tamsayı döndürmeli j. $0 \leq j < e$.
- Bu prosedür birden çok defa çağırılabilir.

Örnek

Şu çağrıya bakalım:

```
Alice(5, 6, [4, 0, 3, 1, 4, 2], [2, 2, 0, 2, 0, 3],
      10, [8, 2, 7, 6, 1, 5, 0, 9, 3, 4])
```

graf şu şekildedir:



ve p başta şudur: $[8, 2, 7, 6, 1, 5, 0, 9, 3, 4]$.

Yukarıdaki değerlere bakarak, permutasyonun optimal skorunun 1 olduğu görülebilir.

Diyelim ki Alice şu 4 hamleyi yapsın:

Argument of t to `Bob`	Return value of `Bob`	Corresponding indices of p	p after the swap by Bob
$[3,1,5,2,0]$	5	$5,2$	$[8,2,5,6,1,7,0,9,3,4]$
$[9,3,7,2,1]$	0	$1,7$	$[8,9,5,6,1,7,0,2,3,4]$
$[5,6,7,8,9]$	1	$5,7$	$[8,9,5,6,1,2,0,7,3,4]$
$[7,5,2,3,6]$	3	$5,2$	$[8,9,2,6,1,5,0,7,3,4]$

Unutmayın ki burada Alice ve Bob her zaman optimal hamleleri yapmıyor olabilir. Bu hamleler amacımızı göstermek için gösterilmiştir. Ayrıca Alice oyunu başlar başlamaz da bitirebilirdi ve bu durumda da skor 1 olurdu.

Alice yukarıdaki hamlelerin hepsini yaptıktan sonra , permutasyonun gerçek skoru 3 ($p[2]=2, p[5]=5, p[7]=7$) olur.

Sonuçta, fonksiyon `Alice()` 1 dönecektir – permutasyonun optimal skoru.

Unutmayın ki Alice 3 skorunu elde etmiş olsa da, `Alice()` fonksiyonunu 1 yerine 3 döndüğünde 0 puan alacaktınız.

Sınırlar

- $2 \le m \le 400$
- $m-1 \le e \le 400$
- $0 \le u[i], v[i] < m$
- $m \le n \le 400$
- $0 \le p[i] < n$
- Graf bağlıdır, self-loops veya multiple edges içermez.
- p bir permutasyondur, yani her $i \ne j$ için $p[i] \ne p[j]$.

Alt Görevler

1. (6 puan) $m = 2$
2. (6 puan) $e > m$
3. (10 puan) $e = m - 1$
4. (24 puan) $e = m = 3$
5. (24 puan) $e = m = 4$
6. (30 puan) $e = m$

Her alt görev için parçalı puan alabilirsiniz. r alt görevin bütün testleri için $\frac{k}{n}$ oranının maksimumu olsun, burada k tur sayısını (yani `Bob()` çağrı sayısı) gösterir. Alt görevin puanı şununla çarpılır:

Şart	Çarpan
$12 \leq r$	0
$3 < r < 12$	$1 - \log_{10}(r-2)$
$r \leq 3$	1

Eğer problemi $3n$ tur içinde çözerseniz, alt görevden tam puan alırsınız. $12n$'den fazla tur kullanmak alt görev için 0 puan almaya sebebiyet verir.(`Output isn't correct` olarak dönüt alırsınız).

Örnek Grader

Örnek grader girdiyi şu formatta okur:

- line 1: $m\ e$
- line $2+i$ $(0 \leq i \leq e-1)$: $u[i]\ v[i]$
- line $2+e$: n
- line $3+e$: $p[0]\ p[1]\ \ldots\ p[n-1]$

Örnek grader çıktıyı şu formatta verir:

- line 1: final permutation p
- line 2: return value of `Alice()`
- line 3: actual score of final permutation
- line 4: the number of turns