

Permutasiya Oyunu (permgame)

Alis və Bob uşaqlıq dostlarıdır və intellektual oyunlar oynamağı sevirlər. Bu gün onlar qraf üzərində yeni bir oyun oynayırlar.

Oyun dəstinə 0-dan $m - 1$ -ə qədər tam ədədlərlə nömrələnmiş m sayda təpədən və 0-dan $e - 1$ -ə qədər tam ədədlərlə nömrələnmiş e sayda kənardan ibarət **əlaqəli** bir qraf daxildir. i -ci kənar $u[i]$ və $v[i]$ təpələrini birləşdirir.

Oyun dəstində həmçinin n uzunluqlu $p[0], p[1], \dots, p[n - 1]$ permutasiyası da var, burada $m \leq n$. Permutasiya 0-dan $n - 1$ -ə qədər bütün ədədlərin yalnız bir dəfə hər hansı sırada yer aldığı massivdir. p permutasiyasının **xalı**, $p[i] = i$ olan indekslərin sayıdır.

Oyun maksimum 10^{100} gedişə qədər davam edir. Hər gedişdə aşağıdakılar baş verir:

1. Əgər Alis oyunu bitirməyə qərar verərsə, oyun dərhal dayanır.
2. Əks halda, Alis **fərqli indekslərdən** ibarət $t[0], t[1], \dots, t[m - 1]$ massivini seçir, burada $0 \leq t[i] < n$. Qeyd edək ki, $t[0] < t[1] < \dots < t[m - 1]$ olması tələb **olunmur**.
3. Bob qrafın kənarlarından birinin indeksini ($0 \leq j < e$) seçir və $p[t[u[j]]]$ ilə $p[t[v[j]]]$ elementlərinin yerini dəyişdirir.

Alis permutasiyanın yekun xalını maksimum etməyə, Bob isə onu minimum etməyə çalışır.

Sizin tapşırığınız Alisə Boba qarşı (Bobun gedişləri qiymətləndirici tərəfindən simulyasiya olunur) oynaması üçün kömək etməkdir.

Optimal xal, Alis və Bob hər ikisi optimal strategiya ilə oynayarsa əldə olunan yekun permutasiyanın xalıdır.

Siz bu optimal xalı tapmalı və sonra Bobla oynayaraq bir neçə gedişdən sonra **ən azı** bu optimal xalı əldə etməlisiniz.

Qeyd edək ki, Alisin strategiyası Bobun etdiyi gedişlərə baxmayaraq, Bobun optimal olmayan gedişlər etdiyi halda belə, həmişə işləməlidir.

İmplementasiya detalları

Aşağıdakı proseduru icra etməlisiniz:

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

- m : qrafdakı təpələrin sayı.
- e : qrafdakı kənarların sayı.
- u və v : qrafın kənarlarını təsvir edən e uzunluqlu massivlər.
- n : permutasiyanın uzunluğu.
- p : permutasiyanı təsvir edən n uzunluqlu massiv.
- Bu prosedur yalnız bir dəfə çağırılır.
- Bu prosedur bir tam ədəd qaytarmalıdır – oyunun optimal xalı.

Bu prosedur daxilində aşağıdakı proseduru çağırmaq olar:

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

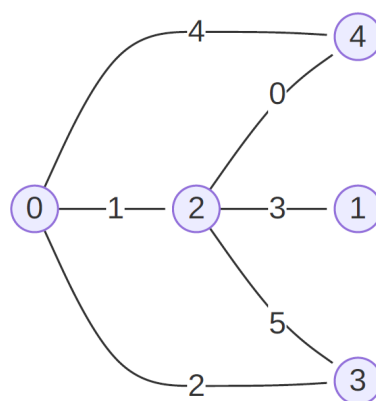
- t : fərqli indeksləri ehtiva edən m ölçülü massiv ($0 \leq t[i] < n$ və $i \neq j$ üçün $t[i] \neq t[j]$).
- Bu prosedur $0 \leq j < e$ şərtini ödəyən bir tam j ədədi qaytarır.
- Bu prosedur bir neçə dəfə çağırıla bilər.

Nümunə

Aşağıdakı çağırışı nəzərdən keçirək:

```
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])
```

Qraf belə görünür:



və p ilkin olaraq $[8, 2, 7, 6, 1, 5, 0, 9, 3, 4]$ -dür.

Yuxarıdakı məhdudiyyətləri nəzərə alaraq, permutasiyanın optimal xalının 1 olduğunu sübut edə bilərik.

Tutaq ki, Alis aşağıdakı 4 gedişi edir:

Bob proseduruna ötürülən t	Bob prosedurundan qayıdan dəyər	p massivində uyğun indekslər	Dəyişmədən sonra p
[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]

Qeyd edək ki, burada Alis və Bobun gedişləri optimal olmaya bilər. Bu gedişlər sadəcə nümunə məqsədilə göstərilib. Onu da qeyd edək ki, Alis oyunu dərhal bitirə də bilərdi, çünki permutasiyanın başlanğıc xalı elə 1-dir.

Alis yuxarıdakı gedişləri etdikdən sonra permutasiyanın faktiki xalı 3 olacaq ($p[2] = 2, p[5] = 5, p[7] = 7$).

Sonda `Alice()` proseduru 1 – permutasiyanın optimal xalını qaytaracaq.

Qeyd edək ki, Alis Bobla oynayaraq 3 xal qazansa da, əgər `Alice()` prosedurundan qaytarılan dəyər 1 əvəzinə 3 olsaydı, 0 bal alacaqdınız.

Məhdudiyyətlər

- $2 \leq m \leq 400$
- $m - 1 \leq e \leq 400$
- $0 \leq u[i], v[i] < m$
- $m \leq n \leq 400$
- $0 \leq p[i] < n$
- Qraf əlaqəlidir, döngülər (self-loop) və paralel kənarlar (multi-edge) yoxdur.
- p permutasiyadır, yəni istənilən $i \neq j$ üçün $p[i] \neq p[j]$.

Alt Tapşırıqlar

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

Hər alt tapşırıq üçün qismən bal ala bilərsiniz. Qoy r alt tapşırıq daxilindəki bütün testlər üzrə $\frac{k}{n}$ nisbətinin maksimumu olsun, burada k gedişlərin (yəni `Bob()` çağırışlarının) sayı, n isə permutasiyanın uzunluğudur. Onda bu alt tapşırıq üzrə balınız aşağıdakı cədvələ əsasən bir ədədə vurulur:

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

Yəni əgər tapşırığı $3n$ gediş daxilində həll etsəniz, həmin alt tapşırıqdan tam bal alırsınız. Əgər $12n$ və ya daha çox gediş etsəniz, həmin alt tapşırıqdan 0 bal verilir (Output isn't correct kimi görünəcək).

Nümunə Qiymətləndirici

Nümunə qiymətləndirici giriş verilənlərini aşağıdakı formatda oxuyur:

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

Nümunə qiymətləndirici çıxışı aşağıdakı formatda çap edir:

- sətir 1: yekun permutasiya p
- sətir 2: `Alice()` prosedurunun qaytardığı dəyər
- sətir 3: yekun permutasiyanın faktiki xalı
- sətir 4: gedişlərin sayı