

Permutatsiya O'yini (permgame)

Alisa va Bob bolalikdan do'st va ular intellektual o'yinlarni juda yaxshi ko'rishadi. Bugun ular graflarda yangi o'yin o'ynashmoqda.

O'yin to'plamida m ta **bog'langan** grafik tugunlari mavjud, ular 0 dan $m - 1$ gacha raqamlangan va e ta qirralar mavjud, ular 0 dan $e - 1$ gacha raqamlangan. i -qirra $u[i]$ va $v[i]$ tugunlarni bog'laydi.

O'yin to'plamida yana uzunligi n bo'lgan permutatsiya $p[0], p[1], \dots, p[n - 1]$ mavjud, bu yerda $m \leq n$. Permutatsiya — bu shunday massivki, unda 0 dan $n - 1$ gacha bo'lgan har bir son aynan bir marta, biror tartibda qatnashgan. Permutatsiyaning **hisobi (score)** — bu $p[i] = i$ bo'lgan indekslar soni.

O'yin ko'pi bilan 10^{100} yurishda davom etadi. Har bir yurishda quyidagilar sodir bo'ladi:

1. Agar Alisa o'yinni tugatishga qaror qilsa, o'yin to'xtaydi.
2. Aks holda, Alisa **turli** indekslar $t[0], t[1], \dots, t[m - 1]$ ni tanlaydi, bu yerda $0 \leq t[i] < n$. E'tibor bering, bunda $t[0] < t[1] < \dots < t[m - 1]$ bo'lishi shart **emas**.
3. Bob grafikdagi qirralardan birini tanlaydi: $0 \leq j < e$, va $p[t[u[j]]]$ bilan $p[t[v[j]]]$ ni qiymatlarini alishtiradi (swap).

Alisa permutatsiyaning yakuniy hisobini maksimal qilishni istaydi, Bob esa bu hisobni minimal qilishni istaydi.

Sizning vazifangiz — Alisaga yordam berish va Bobga qarshi o'ynash (Bobning harakatlari grader tomonidan simulyatsiya qilinadi, ya'ni grader Bob o'rniga o'ynaydi, siz esa Alisa o'rnida).

Keling, *optimal hisob* — bu Alisa va Bob har ikkalasi ham optimal o'ynagandagi permutatsiyaning yakuniy hisobi bo'lsin.

Sizdan quyidagi ikki ish talab qilinadi:

- Permutatsiyaning optimal hisobini aniqlash.
- So'ng, Alisa sifatida o'ynash va bir nechta yurishlardan so'ng **kamida** shu optimal hisobga erishish (ya'ni optimal hisob yoki undan balandroq hisobga erishish).

Eslatma: Alisaning strategiyasi Bob qanday harakat qilmasin, hatto optimal o'ynamasa ham, ishlashi kerak.

Implementation details

Siz quyidagi protsedurani implementatsiya qilishingiz kerak:

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

- m : grafdagi tugunlar soni.
- e : grafdagi qirralar soni.
- u va v : uzunligi e bo'lgan massivlar, qirralarni ifodalaydi.
- n : permutatsiya uzunligi.
- p : uzunligi n bo'lgan massiv, permutatsiyani ifodalaydi.
- Ushbu protsedura aniq bir marta chaqiriladi.
- Ushbu protsedura bitta butun son — o'yin uchun optimal hisobni qaytarishi kerak.

Ushbu protsedura ichida quyidagi yordamchi funksiyani chaqirishingiz mumkin:

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

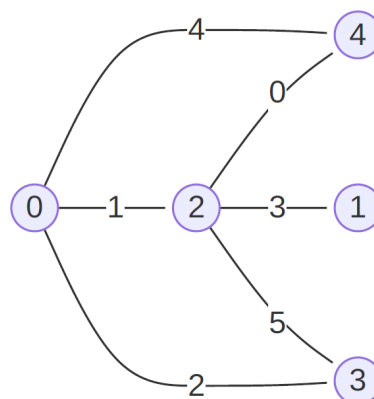
- t : o'lchami m bo'lgan massiv, undagi sonlar o'zaro har hil, ya'ni $0 \leq t[i] < n$ va har qanday $i \neq j$ uchun $t[i] \neq t[j]$.
- Bu funksiya yagona butun son j ni qaytaradi, bu yerda $0 \leq j < e$.
- Ushbu funksiya bir necha marta chaqirilishi mumkin.

Example

Quyidagi chaqiruvni ko'rib chiqing:

```
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 quyidagicha ko'rinishga ega:



va p dastlab $[8, 2, 7, 6, 1, 5, 0, 9, 3, 4]$.

Yuqoridagi cheklovlar asosida, permutatsiyaning optimal hisobi 1 ekanligini isbotlash mumkin.

Tasavvur qilaylik, Alisa quyidagi 4 ta yurishni amalga oshiradi:

Bob ga uzatilgan t argumenti	Bob qaytargan qiymat	p massivining mos indeksleri	Bob almashtirganidan so'ng 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	$[8, 9, 2, 6, 1, 5, 0, 7, 3, 4]$

Eslatma: Alisa va Bob har doim ham optimal yurish qilmasligi mumkin. Bu yurishlar faqat tushuntirish maqsadida ko'rsatilgan. Shuningdek, Alisa o'yinni darhol tugatishi ham mumkin edi, chunki boshlang'ich permutatsiya hisobi allaqachon 1 ga teng edi.

Alisa yuqoridagi barcha yurishlarni bajarganidan so'ng, permutatsiyaning haqiqiy holati $p[2] = 2$, $p[5] = 5$, $p[7] = 7$ bo'lib, umumiy hisob 3 ga teng bo'ladi.

Yakunida `Alice()` funksiyasi 1 — ya'ni permutatsiyaning optimal hisobini qaytaradi.

Eslatma: Garchi Alisa Bob bilan o'ynab hisobni 3 ga yetkazgan bo'lsa-da, agar `Alice()` funksiyasi 3 ni qaytarsa, siz 0 ball olasiz. To'g'ri javob — optimal hisob 1 bo'lishi kerak.

Constraints

- $2 \leq m \leq 400$
- $m - 1 \leq e \leq 400$
- $0 \leq u[i], v[i] < m$
- $m \leq n \leq 400$
- $0 \leq p[i] < n$
- Graf bog'langan bo'lib, o'ziga o'zi tutashgan (self-loop) yoki bir xil ikki uchi bo'lgan bir nechta qirralarga ega emas.
- p — bu permutatsiya, ya'ni $p[i] \neq p[j]$ har qanday $i \neq j$ uchun.

Subtasks

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

6. (30 ball) $e = m$

Har bir subtaskda siz qisman ball olishingiz mumkin. r — bu subtaskdagi barcha testlar orasida $\frac{k}{n}$ nisbatining maksimal qiymati, bu yerda k — yurishlar soni (ya'ni `Bob()` chaqiruvlari soni). Unda sizning subtask bo'yicha ballingiz quyidagi songa ko'paytiriladi:

Shart	Ko'paytuvchi
$12 \leq r$	0
$3 < r < 12$	$1 - \log_{10}(r - 2)$
$r \leq 3$	1

Xususan, agar siz masalani $3n$ yurish ichida yechsangiz, bu subtask uchun to'liq ball olasiz. Agar $12n$ dan ortiq yurish ishlatsangiz, bu subtask uchun 0 ball olasiz (`Output isn't correct` deb ko'rsatiladi).

Sample Grader

Namunviy grader sonlarni quyidagi formatda o'qiydi:

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

Grader quyidagi formatda javoblarni chiqaradi:

- 1-qator: yakuniy permutatsiya p
- 2-qator: `Alice()` funksiyasining qaytargan qiymati
- 3-qator: yakuniy permutatsiyaning haqiqiy hisob qiymati
- 4-qator: yurishlar soni