Diametr (diameter)

Dadorlandtiria mamlakatida n ta shahar bor, ular 0 dan $n-1$ gacha raqamlangan. Shuningdek, mamlakatda aynan n ta to'g'ridan-to'g'ri yo'l barcha shaharlarni bog'lab turadi. Xususan, barcha $0 \leq i \leq n-1$ uchun $u[i]$ va $v[i]$-raqamli shaharlar to'g'ridan-to'g'ri yo'l bilan bog'langan.

Buni qarangki, Dadorlandtiria mamlakatidagi barcha yo'llar ta'mirga muhtoj. Mamlakat rahbari har bir yo'lni ta'mirlash aholi uchun qanchalik noqulaylik tug'dirishini bilmoqchi.

Xususan, har kuni mamlakatda qaysidir yo'l ta'mirlash ishlari uchun yopib qo'yiladi. Noqulaylik darajasi deb, orasidagi sayohat vaqti eng uzoq bo'lgan shaharlar juftligiga aytiladi. E'tibor bering, qaysidir ikki shahar orasida sayohat qilish ilojsiz bo'lib qolishi ham mumkin.

Har bir yo'l uchun, uni yopilishi natijasida mamlakatdagi eng uzoq sayohat vaqtini toping. Agar qaysidir shaharlar orasida sayohat qilish ilojsiz bo'lib qolsa, javob sifatida -1 qaytaring.

Implementation details

Siz quyidagi protsedurani dasturlashingiz kerak:

```
vector<int> find_diameters(int n, vector<int> u, vector<int> v)
```

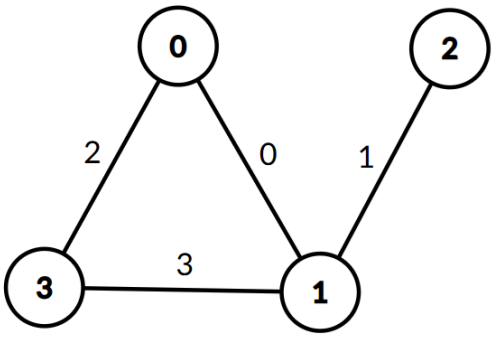
- n: jami shaharlar va yo'llar soni.
- u, v: uzunligi n ga teng bo'lgan massivlar – yo'llar haqida ma'lumot.
- Bu protsedura uzunligi n bo'lgan massiv qaytarishi kerak, bunda i-element i-yo'l yopilishi natijasida hosil bo'ladigan eng uzoq sayohat vaqti, yoki -1.
- Bu protsedura aynan bir marta chaqiriladi.

Example

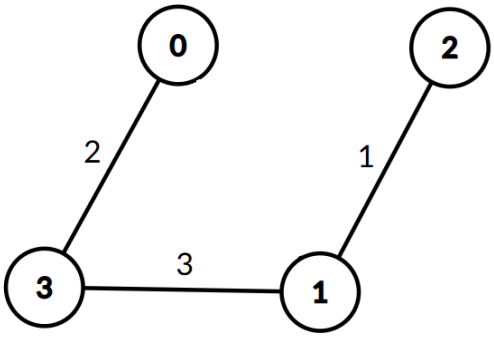
Quyidagi chaqiruvni ko'raylik:

```
find_diameters(4, [1, 1, 0, 3], [0, 2, 3, 1])
```

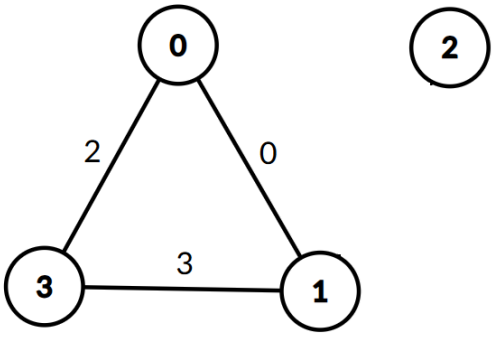
Dastlab, Dadorlandtiria mamlakati quyidagi ko'rinishda:



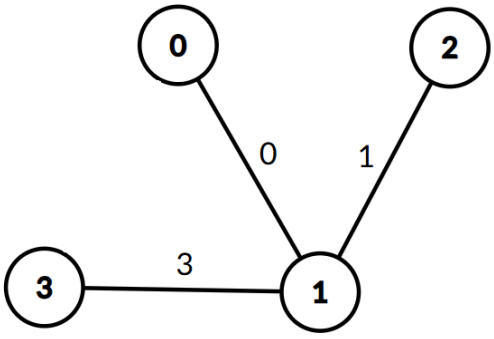
0-yo'lni ta'mirlash paytida, mamlakat quyidagi ko'rinishga keladi, bunda eng uzoq sayohat vaqti 3 ga teng: $0 \to 3 \to 1 \to 2$.



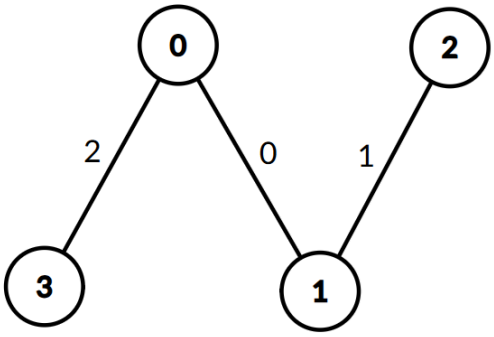
1-yo'lni ta'mirlash paytida, mamlakat quyidagi ko'rinishga keladi, bunda 0 va 2 shaharlar orasida sayohat qilishni iloji qolmaydi.



2-yo'lni ta'mirlash paytida, mamlakat quyidagi ko'rinishga keladi, bunda eng uzoq sayohat vaqti 2 ga teng: $3 \to 1 \to 2$.



3-yo'lni ta'mirlash paytida, mamlakat quyidagi ko'rinishga keladi, bunda eng uzoq sayohat vaqti 3 ga teng: $3 \to 0 \to 1 \to 2$.



Shunday qilib, `find_diameters()` protsedurasi javob sifatida $[3, -1, 2, 3]$ massivini qaytarishi kerak.

Constraints

- $3 \leq n \leq 200\ 000$
- $0 \leq u[i], v[i] \leq n - 1$ va $u[i] \neq v[i]$ (barcha $0 \leq i < n$ uchun)
- Hech qaysi $(u[i], v[i])$ juftligi ikki marta uchramaydi.
- Dastlab, mamlakat **bog'langan**, ya'ni ixtiyoriy shaharlar juftligi uchun ular orasida sayohat qilish mumkin.

Subtasks

1. (3 ball) $u[i] = i$ va $v[i] = (i + 1) \bmod n$ barcha $0 \leq i < n$ uchun
2. (9 ball) $n \leq 100$
3. (21 ball) $n \leq 2000$
4. (9 ball) $k \leq 2000$. Bu yerda k - javob -1 bo'lmaydigan yo'llar soni.
5. (58 ball) Qo'shimcha cheklovlarsiz.

Sample Grader

Namunaviy grader ma'lumotlarni quyidagi tartibda o'qiydi:

- qator 1: n
- qator $2 + i$ $(0 \leq i < n)$: $u[i]$ $v[i]$

Aytaylik, `find_diameters()` protsedurasi javob sifatida r massivni qaytarsin. Namunaviy grader javoblarni quyidagi tartibda chiqaradi:

- qator 1: $r[0]$ $r[1]$ $\ldots$ $r[n-1]$