

Lektvar (The Potion of Great Power)

Kde bolo, tam bolo, bola raz jedna obrovská, až do nebies rastúca stonka fazule. A na tej stonke fazule žilo N šamanov. Každý šaman mal svoje číslo (od 0 do $N - 1$). Domček šamana i sa nachádzal vo výške $H[i]$ nad zemou. Keďže všetky domčeky sú na zvislo rastúcej stonke fazule, vzdialenosť dvoch domčekov je rovná absolútnej hodnote rozdielu ich výšok.

Všetci šamani si svorne nažívali v mieri, až kým sa neukázalo, že medzi sebou majú dvoch zloduchov: *Zlodeja* a *Temného kuchára*.

Zlodej niekde ukradol recept na *Lektvar obrovskej sily*. Zistil ale, že ho sám nevie navariť. Jediný, kto to zvládne, je *Temný kuchár*. *Zlodej* preto potreboval *Temnému kuchárovi* doručiť recept – ale najlepšie tak, aby nik zo zvyšných šamanov nič netušil.

Zlodej si povedal, že mu pri tom určite pomôže, keď zošle na všetkých šamanov *Kliatbu*. Táto *Kliatba* spôsobila, že si šamani prestali navzájom veriť. Presnejšie, vieme, že *Kliatba* fungovala nasledovne:

- V deň 0, keď bola *Kliatba* zoslaná, si všetky dvojice šamanov prestali veriť.
- *Kliatba* je nestabilná. Každú polnoc nastane presne jedna zmena: buď si nejaká dvojica šamanov veriť začne, alebo si nejaká dvojica šamanov veriť prestane.
- Navyše vieme, že vďaka sile kliatby bude platiť, že žiaden šaman nikdy naraz nebude veriť viac ako $D \leq 500$ iným šamanom.

Do dediny šamanov dnes prišiel *Detektív* a presne zistil, ako sa vyvíjala *Kliatba*, teda kto komu kedy začal, resp. prestal veriť. Všetky tieto informácie dostanete ako súčasť vstupu.

Detektív je presvedčený, že *Zlodej* už zvládol *Temnému kuchárovi* oznámiť recept. Navyše zistil, že ich komunikácia musela prebehnúť nasledovne:

1. *Zlodej* (šaman x) prišiel do domu nejakého iného šamana x' , ktorý v danej chvíli *Zlodejovi* veril.
2. V tom istom čase *Temný kuchár* (šaman y) prišiel do domu nejakého iného šamana y' , ktorý v danej chvíli *Temnému kuchárovi* veril.
3. Oba (*Zlodej* aj *Temný kuchár*) sa naraz nenápadne vyklonili z okna a *Zlodej* recept *Temnému kuchárovi* jednoducho nadiktoval.

Šamani x' aj y' mohli byť úplne ľubovoľní, teda až na to, že $x' \neq x$ a $y' \neq y$.

Je dokonca možné (ak si zloduchovia x a y práve navzájom verili), že dom, ktorý navštívil *Zlodej*, mohol patriť *Temnému kuchárovi* a/lebo naopak. (Formálne, je možné, že $y' = x$ a/lebo $x' = y$.)

Je tiež možné, že $x' = y'$: ak existoval niekto, komu obaja zloduchovia verili, mohli obaja navštíviť ten istý dom a vykloniť sa spolu z toho istého okna.

Detektív potrebuje vašu pomoc s vyšetrovaním. Má niekoľko hypotéz. Každá hypotéza je nasledujúceho tvaru: „*Zlodej* je šaman x , *Temný kuchár* je šaman y (pričom $x \neq y$) a komunikácia prebehla v deň v .“

Samozrejme, nenápadne diktovať recept na veľkú vzdialenosť je náročné. Preto platí, že hypotéza je tým menej vierohodná, čím je väčšia vzdialenosť, na ktorú by museli zloduchovia komunikovať.

Pre každú hypotézu preto zistíte, na akú najmenšiu vzdialenosť mohli zloduchovia komunikovať, ak by tá hypotéza bola pravdivá. Formálne, vypočítajte $\min |H[x'] - H[y']|$, kde minimum berieme cez všetkých $x' \neq x$, ktorí v danej chvíli veria x , a cez všetkých $y' \neq y$, ktorí v danej chvíli veria y .

Library

Toto je interaktívna úloha. Detektívove hypotézy je potrebné spracovať online – teda každú hypotézu musíte vyhodnotiť skôr, než sa dozviete nasledujúcu.

Vaše riešenie by malo implementovať funkcie `init`, `curseChanges` a `question`. Grader ich bude volať v nasledovnom poradí:

1. Grader najskôr raz zavolá vašu funkciu `init`, čím vám oznámi, ako vyzerá dedina.
2. Potom raz zavolá funkciu `curseChanges`, čím vám oznámi, ako sa menil efekt *Kliatby*.
3. Na záver postupne Q -krát zavolá vašu funkciu `question` – raz pre každú hypotézu.

Podrobnejší popis týchto funkcií:

- `void init(int N, int D, int H[])`
Týmto volaním vám grader oznámi počet šamanov (N), limit na počet ľudí, ktorým bude každý šaman naraz veriť (D) a výšky domov všetkých šamanov (pole H).
- `void curseChanges(int U, int A[], int B[])`
Týmto volaním vám grader oznámi počet polnoci, ktoré prebehli od zoslania kliatby (U) a všetky zmeny v poradí, v ktorom nastali (polia A a B , obe dĺžky U).
Pre každé i (od 0 po $U - 1$ vrátane) platí, že $A[i]$ a $B[i]$ sú čísla tých dvoch šamanov, ktorých stav sa zmenil na konci dňa i . (Ak si navzájom verili v deň i tak o polnoci si veriť prestali, a ak si doteraz neverili, tak si o polnoci veriť začali.)
- `int question(int x, int y, int v)`
Parametre x , y a v predstavujú číslo *Zlodeja*, číslo *Temného kuchára* a číslo dňa, kedy sa mali rozprávať.

Vaša funkcia by mala vypočítať a vrátiť príslušnú minimálnu vzdialenosť komunikácie, resp. 10^9 , ak komunikácia nemohla prebehnúť (lebo aspoň jeden spomedzi x a y v deň v nikomu neveril).

Upozorňujeme, že ak mohlo nastať $x' = y'$, teda ak niekto v deň v veril x aj y , je správnu návratovou hodnotou 0.

Constraints

$$2 \leq N \leq 10^5$$

$$1 \leq D \leq 500$$

$$\text{pre všetky } i: 0 \leq H[i] \leq 10^9$$

$$0 \leq U \leq 2 \cdot 10^5$$

$$\text{pre všetky } j: 0 \leq A[j], B[j] < N \text{ a } A[j] \neq B[j]$$

$$1 \leq Q \leq 50\,000$$

$$\text{pre každú hypotézu: } 0 \leq x, y < N \text{ a } x \neq y$$

$$\text{pre každú hypotézu: } 0 \leq v \leq U$$

Time limit: 3.0 s

Memory limit: 256 MiB

Examples

```
init(N=6, D=5, H={ 2, 42, 1000, 54, 68, 234 });

// číslo nového dňa: 1 2 3 4 5 6 7 8 9 10 11
curseChanges(U=11, A={ 0, 2, 3, 3, 3, 1, 5, 0, 3, 1, 3 },
              B={ 1, 0, 4, 5, 5, 3, 3, 5, 0, 3, 5 });

question(x=0, y=3, v= 4) returns 26
// vysvetlenie: |H[1]-H[4]| = 26

question(x=3, y=0, v= 8) returns 0
// vysvetlenie: |H[1]-H[1]| = 0 a tiež |H[5]-H[5]| = 0

question(x=0, y=5, v= 5) returns 1000000000
// vysvetlenie: šaman y=5 v deň v=5 nikomu neveril

question(x=3, y=0, v=11) returns 14
// vysvetlenie: |H[4]-H[3]| = 14
```

Na obrázku 1 sú znázornené odpovede pre hypotézy z vyššie uvedeného príkladu.

Na obrázku 2 sú pre každý deň znázornené dvojice šamanov, ktorí si veria.

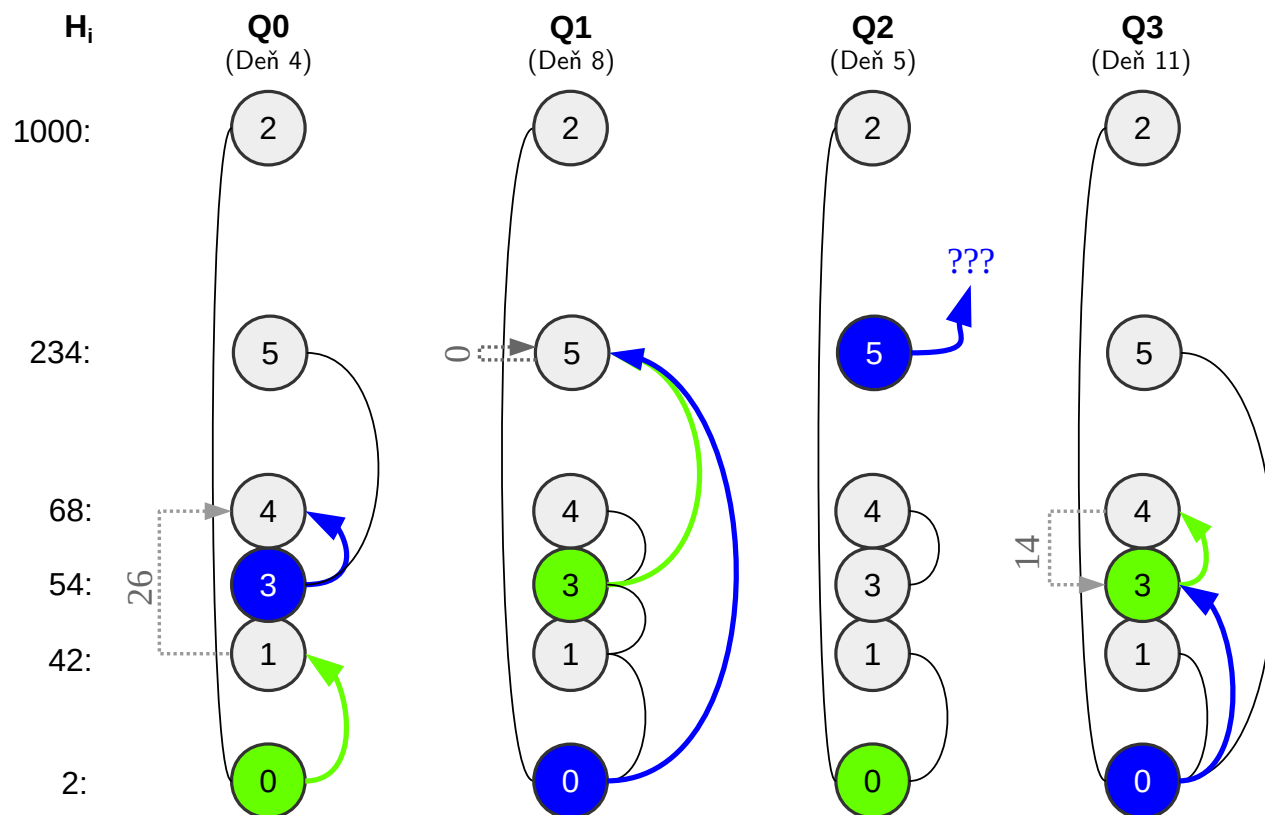
Detailný popis riešenia prvej hypotézy:

V prvej hypotéze je *Zlodejom* šaman $x = 0$, *Temným kuchárom* šaman $y = 3$ a rozprávať sa mali v deň $v = 4$.

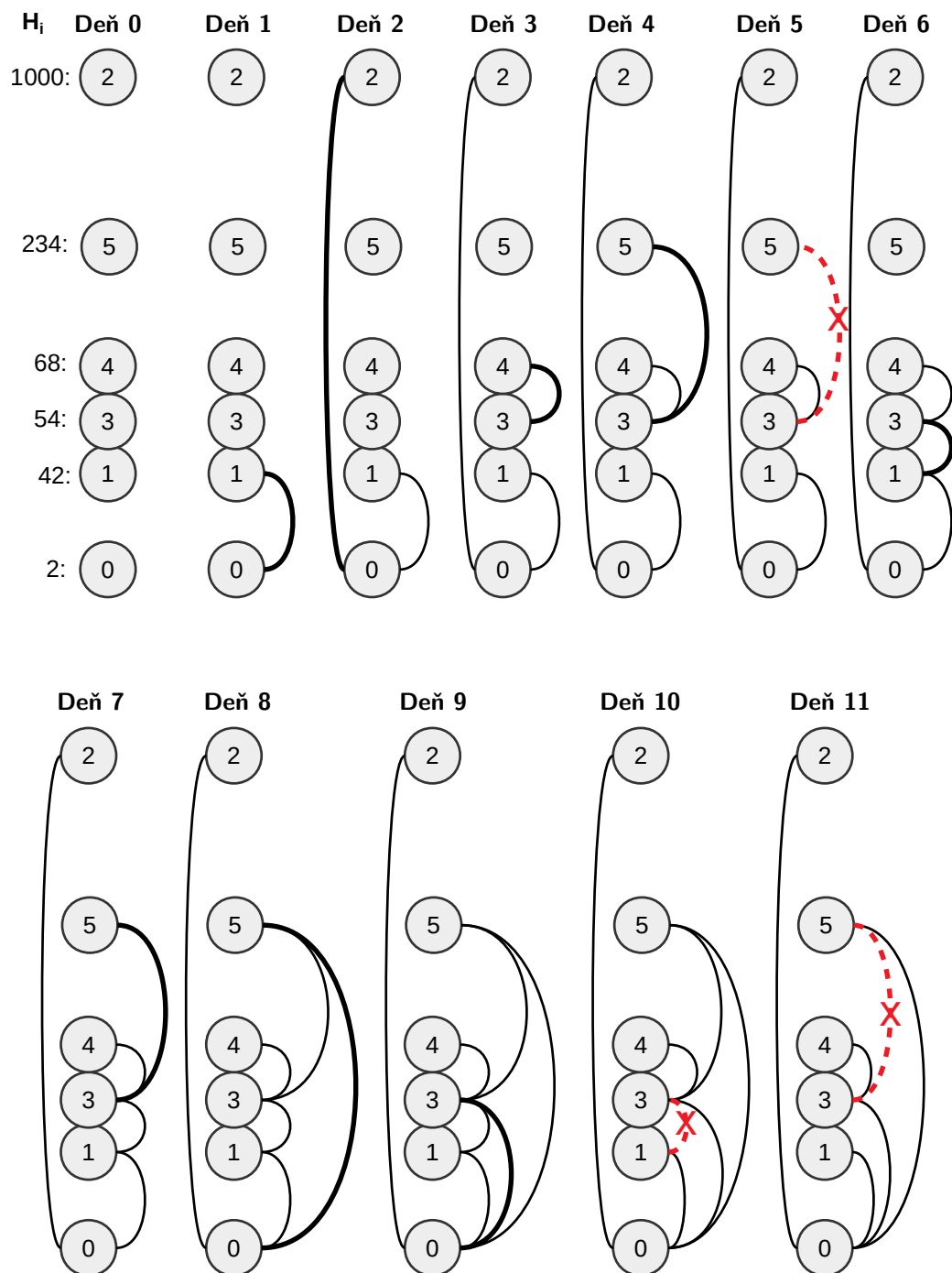
V tento deň šamanovi 0 verili šamani 1 a 2, zatiaľ čo šamanovi 3 verili šamani 4 a 5. Preto sú štyri možnosti, ako mohlo prebehnúť oznámenie receptu. (Značením $x' \rightarrow y'$ myslíme, že recept bol oznámený z domu šamana x' do domu šamana y' .)

- $1 \rightarrow 4$, vzdialenosť: 26,
- $1 \rightarrow 5$, vzdialenosť: 192,
- $2 \rightarrow 4$, vzdialenosť: 932, alebo
- $2 \rightarrow 5$, vzdialenosť: 766.

Správnou odpoveďou je najmenšia z týchto vzdialeností, teda 26.



Obr. 1: Riešenia všetkých hypotéz z príkladu.



Obr. 2: Vývoj toho, ktoré dvojice si v príklade navzájom verili.

Practice

K tejto úlohe si môžete stiahnuť súbor *sample.zip*, ten bude obsahovať nasledovné súbory:

- `grader.cpp`: ukážkový grader. Tento grader spraví volania zodpovedajúce vyššie uvedenému príkladu. Môžete si ho samozrejme upraviť, ak chcete vyskúšať aj iné vstupy.
- `potion.cpp`: kostra riešenia, ktoré máte implementovať.

Grader a vaše riešenie by ste mali kompilovať spolu, napr. takýmto príkazom:

```
g++ -o potion grader.cpp potion.cpp
```

Ak používate IDE, pridajte `grader.cpp` aj `potion.cpp` do toho istého projektu.

Grading

Subtask	Points	Constraints
1	0	sample
2	17	$Q, U \leq 1000$
3	14	v každej otázke platí $v = U$
4	18	pre každého šamana platí $H[i] \in \{0, 1\}$
5	21	$U, N \leq 10\,000$
6	30	bez ďalších obmedzení