

The Potion of Great Power

C'era una volta la *Terra degli Sciamani*, dove tutti vivevano nella *Pianta di Fagioli Celestiale*. Ogni sciamano aveva un numero identificativo univoco i , compreso tra 0 ed $N - 1$, e un valore di altezza H_i , che rappresenta quanto in alto vivono rispetto al suolo. La distanza tra due altezze è il valore assoluto della loro differenza.

Tutti gli sciamani vivevano in pace, finché un giorno uno di essi non rubò la formula della famosa *Pozione del Grande Potere*. Per coprire le proprie tracce, il *Ladro* ha posto una *Maledizione* sulla terra: la maggior parte degli abitanti non si fidano più tra di loro. . .

Cheché la situazione sia avversa, l'*Ordine dei Grandi Investigatori* ha raccolto le seguenti informazioni sulla *Maledizione*:

- quando la *Maledizione* viene posta per la prima volta, tutti smettono di fidarsi di tutti gli altri;
- la *Maledizione* è instabile: alla fine di ogni giorno (esattamente a mezzanotte), una coppia di sciamani inizierà a o terminerà di fidarsi;
- purtroppo, ogni sciamano si fiderà al più di altri D sciamani in ogni dato momento.

Hanno anche ricostruito un rapporto contenente la lista di chi si fidava di chi: per ogni notte sappiamo quale coppia di sciamani ha iniziato/finito di fidarsi.

Credono che il *Ladro* abbia sussurrato la formula ad uno *Sciamano Malvagio*. Per evitare di essere scoperti, entrambi hanno visitato la casa di uno dei loro (rispettivi) amici fidati. Durante la visita, il *Ladro* ha sussurrato la formula allo *Sciamano Malvagio* dalla finestra. (Nota bene: questo amico fidato non doveva necessariamente essere a casa quando questo è successo. Infatti, è addirittura possibile che il *Ladro* e lo *Sciamano Malvagio* hanno visitato le proprie case a vicenda – gli sciamani sono strani.)

Fortunatamente, i sussurri riescono a sentirsi solo a brevi distanze, quindi l'*Ordine* sa che i due amici fidati visitati dal *Ladro* e dallo *Sciamano Malvagio* devono vivere molto vicini tra di loro.

Ti chiedono di aiutarli con la loro investigazione. Vorrebbero testare i loro sospetti: se il *Ladro* fosse x , lo *Sciamano Malvagio* fosse y e la formula fosse stata sussurrata nel giorno v , quale sarebbe la minima distanza che la formula sussurrata dovrebbe percorrere? Ovvero, qual è la distanza minima tra gli appartamenti di due sciamani x' e y' (ovvero $\min(|H_{x'} - H_{y'}|)$) tali che x' è un amico fidato di x e y' è un amico fidato di y nel giorno v ?

Ti verranno condivise tutte le informazioni in possesso dell'*Ordine*, poi ti verranno fatte una serie di domande. È necessario che tu risponda ad ogni domanda immediatamente, prima di ricevere quella dopo.

Libreria

Questo è un problema interattivo. Devi implementare le seguenti funzioni:

- `void init(int N, int D, int H[])`
 N è il numero di sciamani, D è il massimo numero di amici fidati che uno sciamano può avere in ogni istante e H è un vettore di lunghezza N , dove $H[i]$ indica l'altitudine dello sciamano i (per ogni $0 \leq i < N$).
- `void curseChanges(int U, int A[], int B[])`
 U è il numero di giorni, A e B sono vettori di lunghezza U , dove $A[i]$ e $B[i]$ sono la coppia di sciamani che hanno iniziato o smesso di fidarsi l'uno dell'altro alla fine del giorno i (per ogni $0 \leq i < U$). Quindi, se $A[i]$ e $B[i]$ erano amici fidati il giorno i , non lo saranno più il giorno successivo (oppure viceversa).
- `int question(int X, int Y, int V)`
 X è il presunto *Ladro*, Y è il presunto *Sciamano Malvagio* e V è il giorno sospetto. Devi calcolare la minima distanza che la formula che è stata sussurrata avrebbe dovuto percorrere, partendo da un amico fidato X' di X arrivando ad un amico fidato Y' di Y . Nel caso X e Y si fidassero della stessa persona (ovvero $X' = Y'$) devi ritornare 0. Se X o Y non hanno amici fidati, ritorna 10^9 .

Le prime due funzioni verranno invocate una sola volta, nell'ordine in cui appaiono sopra, all'inizio dell'esecuzione del programma. Successivamente, la funzione `question` verrà invocata alcune volte. Il numero di invocazioni è Q .

Assunzioni

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

$$1 \leq D \leq 500$$

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

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

$$0 \leq H_i \leq 10^9 \text{ per ogni } i \text{ (} 0 \leq i < N \text{)}$$

$$0 \leq A[j], B[j], X, Y < N \text{ e } X \neq Y \text{ e } A[j] \neq B[j] \text{ per ogni } j \text{ (} 0 \leq j < U \text{)}$$

$$0 \leq V \leq U$$

Limite di tempo: 3.0 s

Limite di memoria: 256 MiB

Esempi

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

```
// Giorno:          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) -> 26;
// Spiegazione: |H[1]-H[4]| = 26
```

```
question(X=3, Y=0, V= 8) -> 0;
// Spiegazione: |H[1]-H[1]| = 0 o |H[5]-H[5]| = 0
```

```
question(X=0, Y=5, V= 5) -> 1000000000;
// Spiegazione: Y non ha amici fidati nel giorno 5
```

```
question(X=3, Y=0, V=11) -> 14;
// Spiegazione: |H[4]-H[3]| = 14
```

La figura 1 mostra le risposte alle domande nell'esempio precedente, mentre la figura 2 mostra, ogni giorno, le amicizie fidate fra sciamani.

Esempio dettagliato: Per la prima domanda, il *Ladro* è $X = 0$, lo *Sciamano Malvagio* è $Y = 3$ e la formula è stata sussurrata il giorno $V = 4$. Gli amici fidati di X sono 1 e 2, mentre gli amici fidati di Y sono 4 e 5, quindi i possibili percorsi per il sussurro sono:

- $1 \rightarrow 4$, distanza: 26,
- $1 \rightarrow 5$, distanza: 192,
- $2 \rightarrow 4$, distanza: 932, e
- $2 \rightarrow 5$, distanza: 766.

Quindi, la risposta è 26, ovvero la distanza più corta.

Pratica

Puoi scaricare un archivio *sample.zip* contenente i seguenti file:

- `grader.cpp`, un grader d'esempio. Usa l'esempio precedente come input per il tuo programma. Puoi modificarne le costanti per provare altri input.
- `potion.cpp`, una soluzione d'esempio, che dovresti espandere per risolvere il problema.

Dovresti compilare questi due file insieme, ad esempio usando `g++ -o potion grader.cpp potion.cpp`. Se stai usando un IDE, allora dovresti aggiungere sia `grader.cpp` che `potion.cpp` al tuo progetto o lista di sorgenti da compilare.

Punteggi

Subtask	Punti	Assunzioni
1	0	Casi d'esempio
2	17	$Q, U \leq 1000$
3	14	$V = U$ per ogni domanda
4	18	$H[i] \in \{0, 1\}$ per ogni sciamano i
5	21	$U, N \leq 10000$
6	30	Nessuna limitazione aggiuntiva

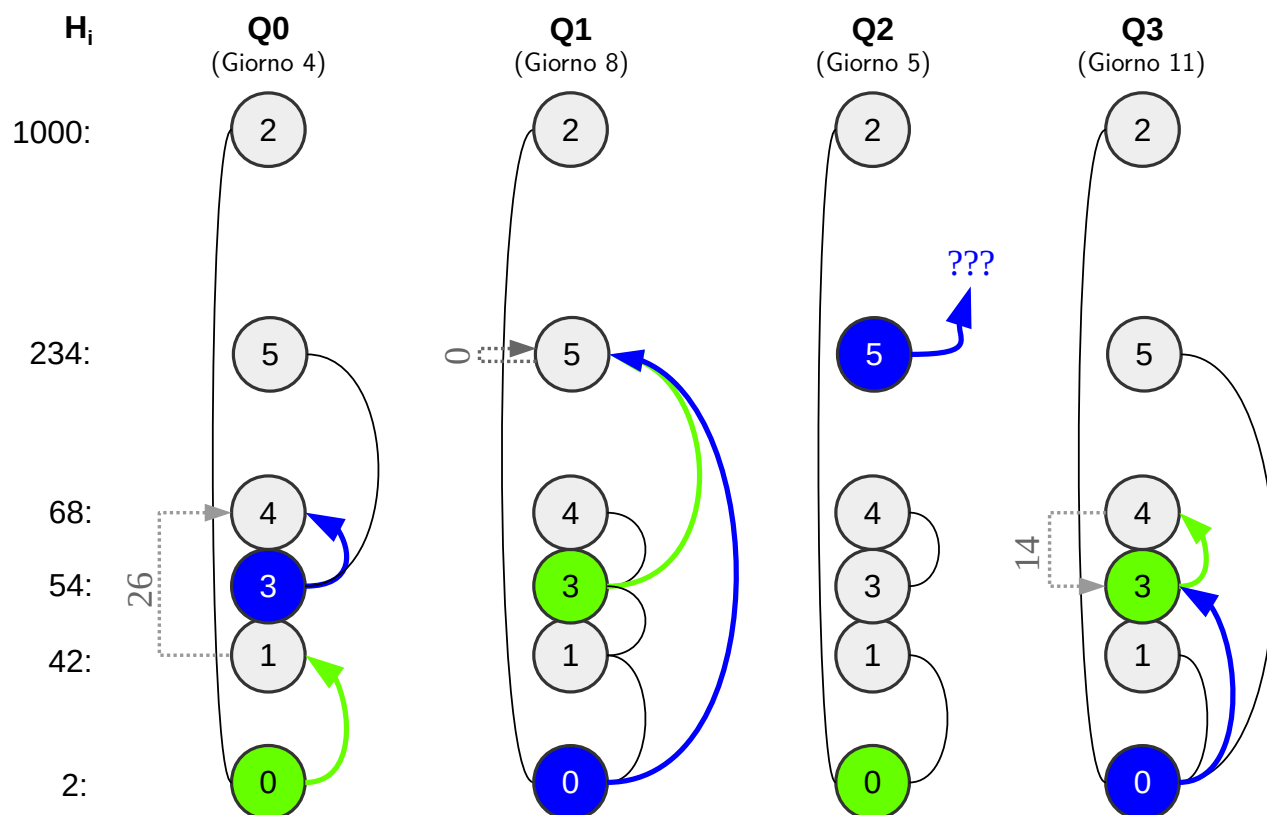


Figura 1: Illustrazione delle quattro domande dell'esempio

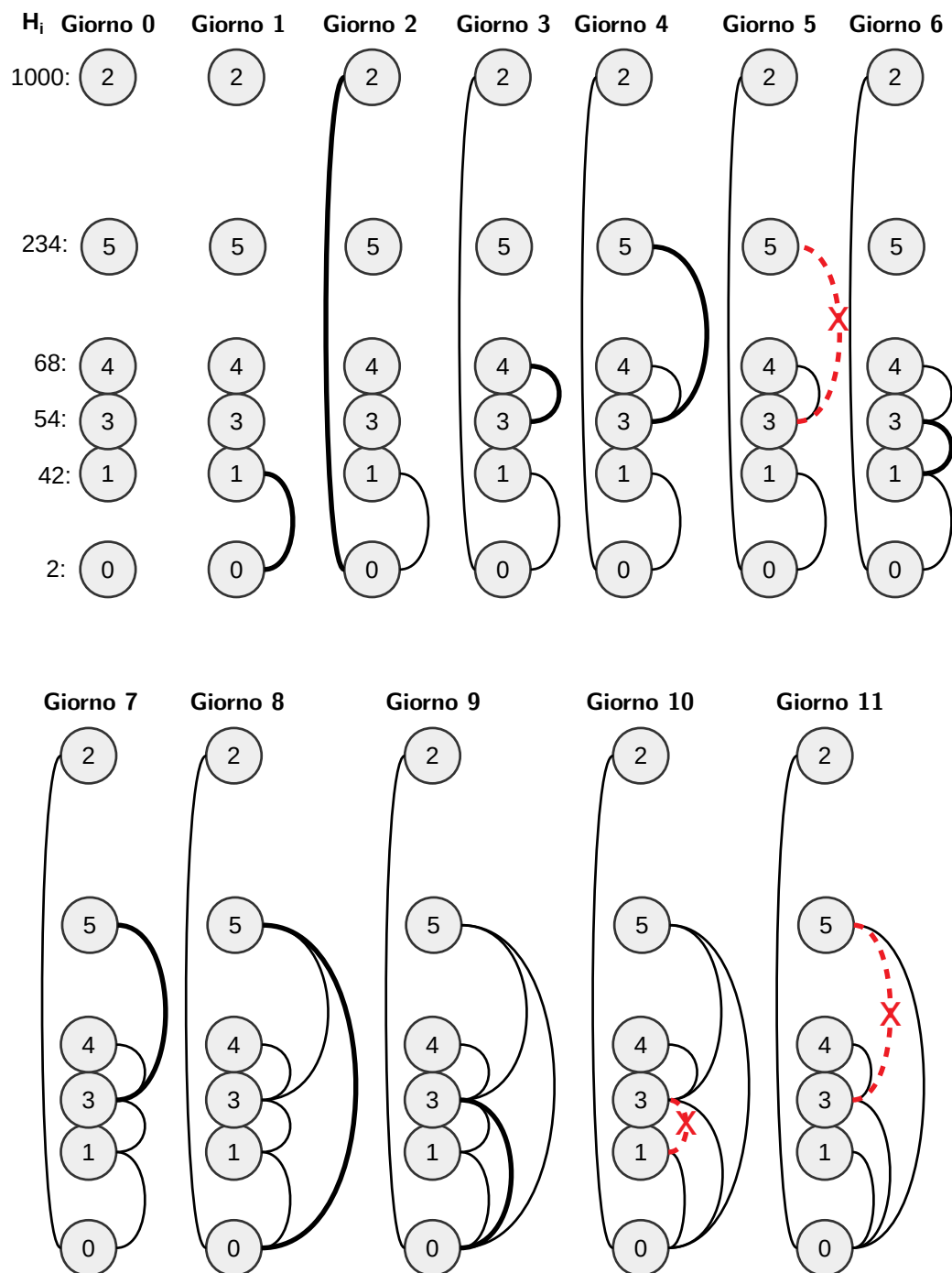


Figura 2: Evoluzione degli amici fidati nell'esempio