

## Der Trank der Macht

Es war einmal, im *Land der Schamaninnen*, die alle auf der *haushohen Bohnenranke* lebten. Jede Schamanin hatte eine sie eindeutig identifizierende Nummer  $x$  zwischen 0 und  $N - 1$  und eine Höhe  $H_i$ , welche angab, wie hoch sie über dem Grund lebte. Die Distanz zwischen zwei Höhen ist gleich deren absoluter Differenz.

Alle Schamaninnen lebten in Frieden zusammen, bis eines Tages eine von ihnen die Formel für den berühmten *Trank der Macht* stahl. Um ihre Spuren zu verwischen, hat die *Diebin* das Land *verflucht*. Die meisten Einwohnerinnen können sich nun nicht mehr gegenseitig vertrauen. . .

Trotz der schweren Umstände hat der *Orden der guten Detektivinnen* folgende Informationen über den *Fluch*:

- Wenn der *Fluch* das erste Mal einsetzt, hören alle auf, anderen zu vertrauen.
- Der *Fluch* ist instabil: Am Ende eines jeden Tages (genau um Mitternacht) wird ein Paar aufhören bzw. beginnen, sich gegenseitig zu vertrauen.
- Bedauerlicherweise wird jede Schamanin zu jedem Zeitpunkt maximal  $D$  anderen vertrauen.

Der Orden hat ausserdem ein Protokoll erstellt, wer wem vertraut: Für jede Nacht ist bekannt, welches Paar an Schamaninnen begonnen/aufgehört hat, sich gegenseitig zu vertrauen.

Der Orden vermutet, dass die *Diebin* die Formel einer *bösen Schamanin* zugeflüstert hat. Um nicht aufzufliegen, haben beide das Haus einer Freundin besucht, welcher sie jeweils vertrauen. Während des Besuchs hat die *Diebin* der *bösen Schamanin* die Formel durch das Fenster zugeflüstert. (Beachte: Es kann sein, dass diese Freundin während des Besuchs nicht zu Hause war. Insbesondere ist es möglich, dass beide das Haus der jeweils anderen besucht haben – Schamanen sind verrückt!)

Glücklicherweise sind Flüstergespräche nur über kurze Distanzen zu hören. Deshalb weiss der Orden, dass die beiden besuchten Häuser (das der *Diebin* und der *bösen Schamanin*) sehr nah aneinander liegen.

Du wirst gebeten bei den Ermittlungen zu helfen. Der Orden würde gerne seine Verdachte überprüfen: Was wäre, wenn die *Diebin*  $x$  war, die *böse Schamanin*  $y$  war und der *Fluch* am Tag  $v$  gefflüstert wurde? Was wäre dann die kürzeste Distanz, die der gefflüsterte *Fluch* zurücklegen müsste? Das heisst, was wäre die kürzeste Distanz zwischen den Appartements zweier Schamaninnen  $x'$  und  $y'$  (also  $\min(|H_{x'} - H_{y'}|)$ ), sodass  $x'$  eine vertraute Freundin von  $x$  und  $y'$  eine vertraute Freundin von  $y$  am Tag  $v$  ist?

Der Orden wird alle seine Informationen mit dir teilen und dir danach eine Reihe an Fragen stellen. Du musst jede Frage sofort beantworten, bevor du die nächste erhältst.

## Bibliothek

Dies ist eine interaktive Aufgabe. Du sollst folgende Funktionen implementieren:

- `void init(int N, int D, int H[])`  
 $N$  ist die Anzahl der Schamaninnen,  $D$  ist die maximale Anzahl Freundinnen, denen eine Schamanin gleichzeitig vertraut, und  $H$  ist ein Array der Länge  $N$ , wobei  $H[i]$  die Höhe der  $i$ -ten Schamanin (für  $0 \leq i < N$ ) ist.
- `void curseChanges(int U, int A[], int B[])`  
 $U$  ist die Anzahl Tagen und  $A$  und  $B$  sind Arrays der Länge  $U$ , sodass  $A[i]$  und  $B[i]$  die Paare an Schamaninnen angibt, welche begonnen oder aufgehört haben, sich am Ende des  $i$ -ten (für  $0 \leq i < U$ ) Tages gegenseitig zu vertrauen. Das heisst, falls sich  $A[i]$  und  $B[i]$  gegenseitig am Tag  $i$  vertrauen, dann vertrauen sie sich am Tag  $i + 1$  nicht mehr, oder umgekehrt.
- `int question(int X, int Y, int V)`  
 $X$  ist die verdächtige *Diebin*,  $Y$  ist die verdächtige *böse Schamanin* und  $V$  ist der verdächtige Tag.  
Du sollst die minimale Distanz zurückgeben, welche der geflüsterte *Fluch* von einer vertrauten Freundin  $X'$  von  $X$  zu einer vertrauten Freundin  $Y'$  zu  $Y$  zurücklegen hätte müssen.  
*Gib 0 zurück, falls jemand sowohl  $X$  als auch  $Y$  vertraut (also  $X' = Y'$ ).*  
*Gib  $10^9$  zurück, falls  $X$  oder  $Y$  keine vertrauten Freundinnen haben.*

Die ersten beiden Funktionen werden zu Beginn der Ausführung jeweils genau einmal in obiger Reihenfolge aufgerufen. Anschliessend wird die Funktion `question` mehrfach aufgerufen. Die Anzahl der Funktionsaufrufe nennen wir  $Q$ .

## Limits

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

$$1 \leq D \leq 500$$

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

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

$$0 \leq H_i \leq 10^9$$

$$0 \leq A[i], B[i], X, Y < N$$

$$X \neq Y \text{ und } A[i] \neq B[i]$$

$$0 \leq V \leq U$$

**Zeitlimit:** 3.0 s

**Speicherlimit:** 256 MiB

## Beispiele

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

```
//Tag:          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;
```

```
// Erklärung: |H[1]-H[4]| = 26
```

```
question(X=3, Y=0, V=8) returns 0;
```

```
// Erklärung: |H[1]-H[1]| = 0 oder |H[5]-H[5]| = 0
```

```
question(X=0, Y=5, V=5) returns 1000000000;
```

```
// Erklärung: Y hat keine Freundinnen an Tag 5.
```

```
question(X=3, Y=0, V=11) returns 14;
```

```
// Erklärung: |H[4]-H[3]| = 14
```

Abbildung 1 verdeutlicht die Antworten auf die Fragen in dem obigen Beispiel und Abbildung 2 zeigt die vertrauten Freundschaften an jedem Tag.

**Detaillierte Beschreibung:** Für die erste Frage ist die *Diebin*  $X = 0$ , die *böse Schamanin* ist  $Y = 3$  und die Formel wurde am Tag  $V = 4$  zugeflüstert. Die vertrauten Freundinnen von  $X$  sind 1 und 2 und die vertrauten Freundinnen von  $Y$  sind 4 und 5. Die möglichen Wege für das Flüstergespräch sind folglich:

- $1 \rightarrow 4$ , Distanz: 26,
- $1 \rightarrow 5$ , Distanz: 192,
- $2 \rightarrow 4$ , Distanz: 932 und
- $2 \rightarrow 5$ , Distanz: 766.

Die Antwort ist 26, da dies die kürzeste Distanz ist.

## Lokales Testen

Du kannst eine Datei *sample.zip* herunterladen, die die folgenden Dateien enthält:

- **grader.cpp**, ein Beispielgrader. Dieser führt dein Programm mit dem obigen Beispiel als Eingabe aus, aber du kannst die Konstanten modifizieren um weitere Eingaben auszuprobieren.
- **potion.cpp**, eine Beispiellösung, welche du erweitern solltest, um diese Aufgabe zu lösen.

Verwende am besten `g++ -o potion grader.cpp potion.cpp` um deine Lösung zu kompilieren (beide Dateien sollten zusammen kompiliert werden).

Falls du eine IDE verwendest, solltest du sowohl `grader.cpp` als auch `potion.cpp` zu deinem Projekt oder der Liste an Quelldateien zum Kompilieren hinzufügen.

## Bewertung

Teilaufgabe	Punkte	Limits
1	0	Beipieltestfall.
2	17	$Q, U \leq 1000$ .
3	14	$V = U$ für alle Fragen.
4	18	$H_i \in \{0, 1\}$ für alle Schamaninnen $i$ .
5	21	$U, N \leq 10000$ .
6	30	Keine weiteren Beschränkungen.

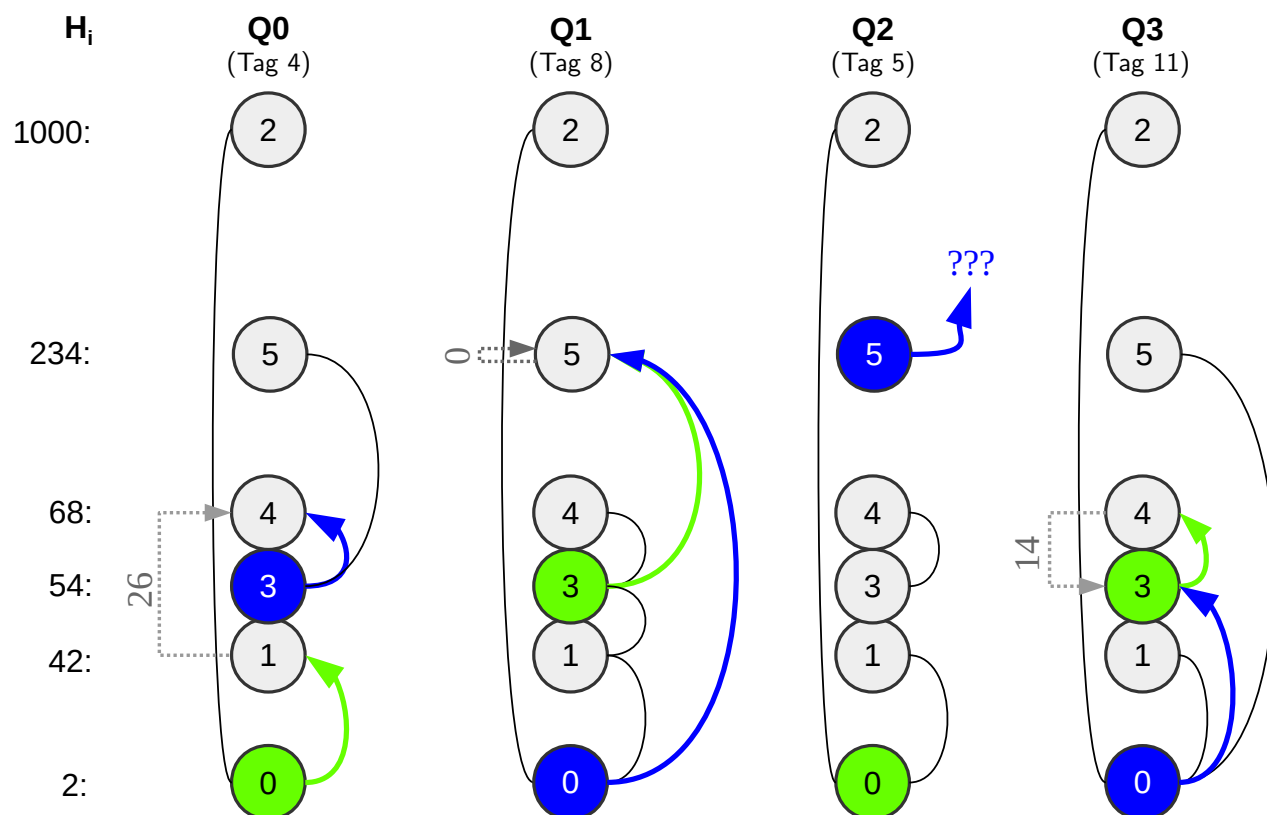


Abbildung 1: Die vier Fragen aus dem Beispiel.

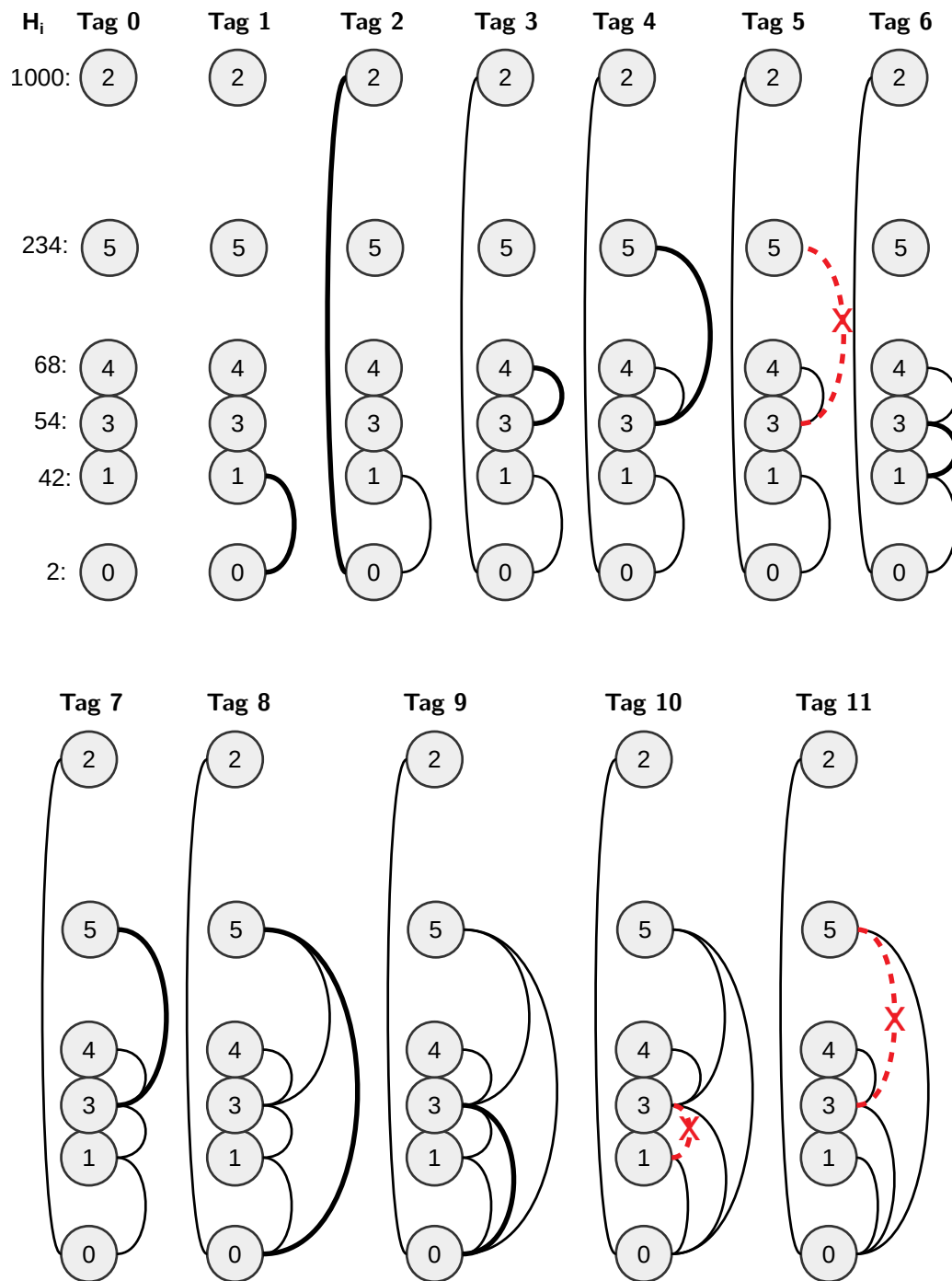


Abbildung 2: Entwicklung der vertrauten Freundschaften in dem Beispiel.