International Olympiad in Informatics 2013
6-13 July 2013
Brisbane, Australia

dreaming

Deutsch — 1.0

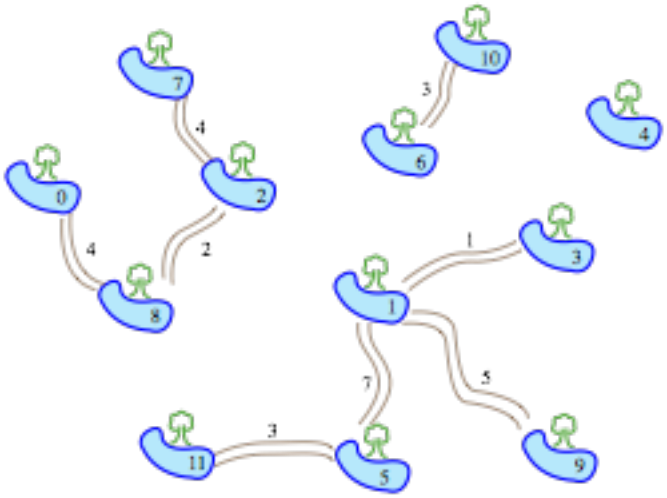
Diese Geschichte fand vor langer Zeit statt, als die Welt noch neu war und die IOI nicht einmal im Traum existierte.

Die Schlange lebt in einem Land mit N Wasserlöchern, nummeriert mit $0, \ldots, N-1$. Es gibt M *Wegstücke*, die jeweils ein Paar von Wasserlöchern miteinander verbinden. Auf diesen kann die Schlange in beide Richtungen reisen. Jedes Paar von Wasserlöchern wird durch höchstens eine Folge von Wegstücken verbunden, jedoch sind möglicherweise manche Paare überhaupt nicht verbunden (deshalb ist $M \leq N-1$). Die Schlange benötigt für die Reise auf einem Wegstück eine bestimmte Anzahl von Tagen: diese Zahl ist möglicherweise für jedes Wegstück anders.

Das Känguru, ein Freund der Schlange, wünscht sich $N - M - 1$ neue Wegstücke zu erstellen, damit es für die Schlange möglich ist, zwischen jedem Paar von Wasserlöchern zu reisen. Das Känguru kann Wegstücke zwischen beliebigen Paaren von Wasserlöchern erstellen. Für jedes Wegstück, welches das Känguru erstellt, wird die Schlange L Tage benötigen.

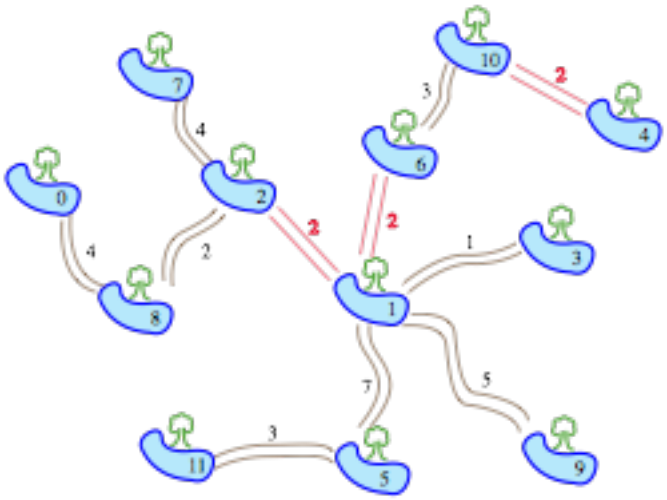
Zusätzlich möchte das Känguru die Reisen der Schlange möglichst schnell machen. Das Känguru wird die neuen Wegstücke so erstellen, dass die längste Reisedauer (engl.: travel time) zwischen je zwei Wasserlöchern so klein wie möglich ist. Hilf dem Känguru und der Schlange die längste Reisedauer zwischen je zwei Wasserlöchern festzustellen, nachdem das Känguru die neuen Wegstücke in der oben beschriebenen Weise hinzugefügt hat.

Beispiel



In der obigen Abbildung sind `N = 12` Wasserlöcher und `M = 8` Wegstücke. Angenommen `L = 2`, dann benötigt die Schlange für jedes neue Wegstück 2 Tage. Das Känguru könnte folgende 3 neue Wegstücke erstellen:

- zwischen den Wasserlöchern 1 und 2;
- zwischen den Wasserlöchern 1 und 6;
- zwischen den Wasserlöchern 4 und 10.



In der obigen Abbildung sieht man die endgültige Menge von Wegstücken. Die längste Reisedauer zwischen je zwei Wasserlöchern ist 18 Tage. Diese ist zwischen den Wasserlöchern 0 und 11. Das ist das kleinstmöglichste Ergebnis—egal wie das Känguru die Wegstücke erstellt, es wird immer ein Paar von Wasserlöchern geben, zwischen denen die Schlange 18 Tage oder mehr reisen muss.

Implementierung

Du musst eine Datei hochladen, welche die Funktion `travelTime()` wie folgt implementiert:

Deine Funktion: `travelTime()`

C/C++

```
int travelTime(int N, int M, int L,
        int A[], int B[], int T[]);
```

Pascal

```
function travelTime(N, M, L : LongInt;
        var A, B, T : array of LongInt) : LongInt;
```

Beschreibung

Diese Funktion soll die grösste Reisedauer (in Tagen) zwischen irgendeinem Paar von Wasserlöchern berechnen, angenommen das Känguru erstellt die `N - M - 1` Wegstücke in einer Weise, sodass alle Wasserlöcher verbunden sind und die grösste Reisedauer so klein wie möglich ist.

Parameter

- `N` : Die Anzahl der Wasserlöcher.
- `M` : Die Anzahl der ursprünglich existierenden Wegstücke.
- `L` : Die Zeit in Tagen, welche die Schlange für ein neues Wegstück benötigt.
- `A`, `B` und `T` : Arrays der Länge `M` für die Endpunkte und Dauer jedes urspünglich vorhandenen Wegstücks, sodass das `i` te Wegstück die Wasserlöcher `A[i-1]` und `B[i-1]` verbindet, und die Schlange `T[i-1]` Tage für die Reise benötigt.
- *Rückgabe*: Die grösste Reisedauer zwischen je zwei Wasserlöchern, wie oben beschrieben.

Beispiel Session

Die folgende Session beschreibt das gegebene Beispiel:

Parameter	Value
N	`12`
M	`8`
L	`2`
A	`[0, 8, 2, 5, 5, 1, 1, 10]`
B	`[8, 2, 7, 11, 1, 3, 9, 6]`
T	`[4, 2, 4, 3, 7, 1, 5, 3]`
Returns	`18`

Beschränkungen

- Zeitlimit: 1 Sekunde
- Speicherlimit: 64 MiB
- $1 \le N \le 100.000$
- $0 \le M \le N - 1$
- $0 \le A[i], B[i] \le N - 1$
- $1 \le T[i] \le 10.000$
- $1 \le L \le 10.000$

Subtasks

Subtask	Punkte	Zusätzliche Beschränkungen
1	14	$M = N - 2$, und von jedem Wasserloch führen genau ein oder zwei ursprünglich vorhandene Wegstücke weg. In anderen Worten: es gibt zwei Mengen von zusammenhängenden Wasserlöchern, und in jeder dieser Mengen bilden die Wegstücke eine nicht verzweigende Folge von Wegstücken.
2	10	$M = N - 2$ und $N \le 100$
3	23	$M = N - 2$
4	18	Von jedem Wasserloch führt höchstens ein ursprünglich vorhandenes Wegstück weg.
5	12	$N \le 3,000$
6	23	*(keine)*

Testen

Der Beispiel-Grader auf deinem Computer liest von der Eingabedatei `dreaming.in`, die das folgende Format haben muss:

- Zeile 1: `N M L`
- Zeilen 2, …, `M + 1`: `A[i] B[i] T[i]`

Zum Beispiel soll das gegebene Beispiel in folgendem Format sein:

```
12 8 2
0 8 4
8 2 2
2 7 4
5 11 3
5 1 7
1 3 1
1 9 5
10 6 3
```

Sprachspezifische Notizen

C/C++	Du musst `#include "dreaming.h"` verwenden.
Pascal	Du musst `unit Dreaming` definieren. Alle Arrays sind mit `0` beginnend indiziert (nicht `1`).

Siehe Lösungsvorlagen auf deiner Maschine für Beispiele.