

Trains

Betrachte ein Zugnetzwerk bestehend aus N Städten. Es gibt genau $N - 1$ direkte Verbindungen zwischen einigen der N Städte, so dass das Zugnetz die Struktur eines Baums mit der Wurzel in Stadt 1 hat. Für jede Stadt i ausser 1 wird dir der Vorgänger von i , nämlich F_i gegeben.

Jede Sekunde, anfangend mit Sekunde 0, verlässt ein Zug die Stadt 1 auf dem kürzesten Pfad zu einer anderen Stadt. Alle Züge haben dieselbe Geschwindigkeit und die direkten Verbindungen sind so, dass es genau eine Sekunde braucht um von einer Stadt zur nächsten zu gelangen, wenn sie eine direkte Verbindung haben.

Für jede Stadt, kennst du Zeit T_i zu der spätestens der erste Zug ankommen muss. (Dies wird von den Bahnreisenden so verlangt).

Deine Aufgabe ist es, Maus Stofl, dem Zugsverantwortlichen, zu helfen herauszufinden ob es einen Weg gibt, die Züge so zu koordinieren, dass für jede Stadt i mindestens ein Zug vor T_i ankommt oder dort durchfährt (der Endbahnhof der Züge ist irrelevant).

Du sollst eine Funktion namens `solution` implementieren, die folgende Argumente hat:

- `int N`: die Anzahl der Städte
- `int *F`: ein 0-basiertes Array mit $N + 1$ Elementen (`F[0]` und `F[1]` sind undefined) welches die Vorgänger darstellt
- `int *T`: ein 0-basiertes Array mit $N + 1$ Elementen (`T[0]` ist undefined), die Wünsche der Bahnkunden.

Die Funktion sollte den `char : 1` zurückgeben, falls die Wünsche aller Bahnkunden zufriedengestellt werden können und 0 sonst.

Eingabe

Eine Vorlage für die Aufgabe liest die Eingabe im Binärformat. Du solltest diesen Teil nicht bearbeiten. (Und es sollte dir auch nicht möglich sein, dies zu tun.)

Ausgabe

Die Vorlage ruft deine Funktion mehrere Male im selben Testfall auf, jedes Mal mit einem möglicherweise verschiedenen Zugnetz. Deine Funktion sollte jedes Mal die richtige Antwort zurückgeben.

Einschränkungen

- Die Funktion wird maximal 50 Mal aufgerufen.
- $1 \leq N \leq 1.5 \cdot 10^5$
- $1 \leq S \leq 5 \cdot 10^6$, S = Summe aller N innerhalb eines Testfalles
- $0 \leq T_i \leq 10^9$ für jedes $1 \leq i \leq N$

Testen

Dies ist eine interaktive Aufgabe. Du musst nichts von der Standarteingabe lesen oder auf die Standartausgabe schreiben.

Der von uns bereitgestellte Code liest von der Standart Eingabe in folgendem Format:

- Die erste Zeile enthält C , die Anzahl wie oft die Funktion `solution` aufgerufen wird.
- C Blöcke bestehend aus ähnlichen Daten, die die Parameter der untenstehenden Funktion beschreiben. Jeder Block hat folgendes Format:

```

1  N
2  F[1] F[2] F[3] ... F[N]
3  T[2] T[3] T[4] ... T[N]
4

```

- Beachte, dass $F[0]$, $F[1]$ and $T[0]$ nicht in der Eingabe erscheinen, da sie undefiniert sind. Du solltest diese Werte nicht benutzen, wenn du die Aufgabe löst.
- Zwei gleichwertige Beispiele werden gegeben: Eines ist im Textformat und folgt der oben beschriebenen Struktur, das andere ist im Binärformat und enthält die selbe Information in einer anderen Codierung.

Teilaufgaben

Es gibt verschiedene Testgruppen, die unabhängig voneinander Punkte geben.

Teilaufgabe	Punkte	Zusätzliche Einschränkungen
1	10	$N \leq 50$
2	10	$50 < N \leq 10^3$
3	20	$10^3 < N \leq 10^4$
4	30	$10^4 < N \leq 5 \cdot 10^4$
5	30	keine

Beispiele

Eingabeparameter	Rückgabewert
$N = 5$ $F = - - 1 2 1 4$ $T = - 10 6 3 5 2$	1
$N = 5$ $F = - - 1 2 1 4$ $T = - 10 1 3 5 2$	0

