

Trains

Имеется сеть железных дорог, соединяющая N городов. В сети существует ровно $N - 1$ направленных дорог между некоторыми парами городов. Сеть образует структуру корневого дерева с корнем в городе 1. Для каждого города, кроме первого, вам дано число F_i — номер родителя вершины i , дороги направлены от родителя к вершине.

Каждую секунду, начиная с 0, один поезд отправляется из города 1 по кратчайшему маршруту в некоторый другой город. Все поезда имеют одинаковую скорость и проезжают любую дорогу между соседними городами за 1 секунду.

Для каждого города i , чтобы выполнить заказы покупателей в нём, первый поезд обязан прибыть туда не позже заданного для этого города времени T_i .

Вы должны помочь диспетчеру и определить, существует ли такой способ составить расписание отправления поездов, чтобы хотя бы один из них прибывал или проезжал через город i до момента времени T_i включительно (пункт назначения поезда не важен).

Вы должны реализовать функцию `solution`, принимающую в качестве аргументов:

- `int N`: количество городов;
- `int *F`: массив из $N + 1$ элементов, проиндексированный с нуля (`F[0]` и `F[1]` не определены), задающий сеть городов;
- `int *T`: массив из $N + 1$ элементов, проиндексированный с нуля (`T[0]` не определено), задающий заказы покупателей.

Функция должна возвращать `int`: 1, если все покупатели будут удовлетворены, и 0 иначе.

Standard input

Вам предоставлен шаблонный код. Шаблонный код считывает входные данные в текстовом или бинарном формате, вы не должны (и не можете) редактировать эту часть кода.

Standard output

Шаблонный код вызывает вашу функцию множество раз во время одного теста, каждый раз, возможно, на разных сетях дорог. Каждый раз функция должна возвращать правильный ответ.

Constraints and notes

- Функция будет вызвана максимум 50 раз.
- $1 \leq N \leq 1.5 \cdot 10^5$
- $1 \leq S \leq 5 \cdot 10^6$, S = сумма всех N в одном тесте
- $0 \leq T_i \leq 10^9$ для всех $1 \leq i \leq N$

Testing

Вам не требуется самостоятельно читать из стандартного ввода или выводить в стандартный вывод.

Предоставленный шаблонный код может считывать данные в следующем текстовом формате:

- Первая строка содержит целое число `C` , количество вызовов функции `solution` .
- Далее следуют `C` блоков данных, которые описывают параметры функции следующим образом:

```
1 N
2 F[1] F[2] F[3] ... F[N]
3 T[2] T[3] T[4] ... T[N]
4
```
- Обратите внимание, что `F[0]` , `F[1]` и `T[0]` не задаются во вводе, их значения не определены, ваша программа не должна пытаться к ним обращаться.
- При запуске на примерах из условия программа запускается на двух эквивалентных тестах, один раз тест считывается в текстовом формате, второй раз в бинарном. Если вы не исправляли шаблонный код, то программа должна корректно работать в обоих случаях.

Subtasks

Тесты будут оцениваться **независимо**.

Subtask	Percentage of test cases	Additional input constraints
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%	none

Examples

Parameters	Return value
<pre>N = 5 F = - - 1 2 1 4 T = - 10 6 3 5 2</pre>	1
<pre>N = 5 F = - - 1 2 1 4 T = - 10 1 3 5 2</pre>	0

