

Trains

Considera una rete ferroviaria che connette N città. Ci sono esattamente $N - 1$ percorsi che collegano direttamente due città in modo che la struttura che si forma sia un albero radicato nella città 1 . Per ogni città i eccetto la 1 ti viene fornito F_i , il padre immediato del nodo i .

Ogni secondo, partendo dal secondo 0 , un treno lascia la città 1 e segue il percorso più breve verso una qualche città. Tutti i treni viaggiano alla stessa velocità e tutti i collegamenti sono lunghi esattamente un secondo.

Per ogni città conosci il tempo T_i entro il quale il primo treno deve arrivare o transitare per la città i , così come richiesto dai clienti.

Devi aiutare l'organizzatore e determinare se esiste un modo per far partire i treni in modo tale che almeno uno di essi arrivi o passi per la città i al massimo al secondo T_i (la destinazione finale di un treno è irrilevante).

Devi implementare una funzione chiamata `solution` che prende i seguenti parametri:

- `int N`: il numero di città;
- `int *F`: a 0 un array indicizzato da 0 con $N + 1$ elementi (`F[0]` e `F[1]` sono indefiniti) che rappresenta la mappa ferroviaria;
- `int *T`: a 0 un array indicizzato da 0 con $N + 1$ elementi (`T[0]` è indefinito) che rappresenta le richieste dei clienti.

La funzione deve ritornare un `char`: `1` se tutti i clienti possono essere soddisfatti, `0` altrimenti.

Standard input

Il codice del grader legge l'input in formato binario. Non devi (e non dovresti poter) modificare questa parte del codice.

Standard output

Il grader chiamerà la funzione più volte durante lo stesso testcase, ogni volta con parametri differenti della rete ferroviaria. Ogni volta la funzione deve ritornare la risposta corretta.

Limiti e note

- La funzione verrà chiamata al più 50 volte.
- $1 \leq N \leq 1.5 \cdot 10^5$
- $1 \leq S \leq 5 \cdot 10^6$, S = somma di tutti gli N in un testcase.
- $0 \leq T_i \leq 10^9$ per ogni $1 \leq i \leq N$

Testing

This is an interactive problem. You don't need to read or write anything from standard input or to standard output.

The code we provide reads from the standard input in the following format:

- The first line contains `C`, the number of times the function `solution` is called.
- `C` similar blocks of data, describing the parameters of the function follow. Each have the format:

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

4

- Note that $F[0]$, $F[1]$ and $T[0]$ don't appear in the input as they are undefined and you should not use their values when solving the problem.
- Two equivalent examples are provided: one of them is in text format and follows the structure described above, the other is in binary format and contains the same information encoded differently.

Subtasks

I testcase verranno valutati **individualmente**.

Subtask	Percentuale dei testcase	Limiti aggiuntivi dell'input
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%	nessuno

Examples

Parametri	Valore di ritorno
$N = 5$ $F = - - 1 2 1 4$ $T = - 10 6 3 5 2$	1
$N = 5$ $F = - - 1 2 1 4$ $T = - 10 1 3 5 2$	0

