

Trains

Razmotrimo železničku mrežu koja pokriva N gradova. Postoji tačno $N - 1$ direktnih pravaca između nekih parova među N gradova tako da ove rute formiraju strukturu ukorenjenog drveta sa korenom koji predstavlja grad čija numeracija je 1. Za svaki grad i osim 1 dat Vam je F_i tj. neposredni roditelj čvora i .

Svake sekunde, počevši nulte sekunde, voz napušta grad 1 na najkraćem putu do nekog drugog grada. Svi vozovi imaju istu brzinu i svi direktni pravci su takvi da je potrebno tačno 1 sekunda da se obavi transfer između dva grada koja dele direktnu vezu.

Za svaki grad znate poslednji vremenski trenutak T_i kada prvi voz mora stići u grad, jer tako zahtevaju korisnici železničkih usluga.

Morate pomoći dispečeru voza i utvrditi da li postoji način za usmeravanje svakog voza tako da najmanje jedan od njih stigne u grad i ili prođe kroz grad i pre vremenskog trenutka T_i ili tačno u vremenski trenutak T_i (krajnje destinacije vozova nisu bitne).

Morate implementirati funkciju koja se zove `solution` koja kao argumente uzima:

- `int N`: broj gradova;
- `int *F`: niz koji počinje indeksom 0 i koji ima $N + 1$ elemenata (`F[0]` i `F[1]` nisu definisane) koji predstavlja mapu;
- `int *T`: niz koji počinje indeksom 0 i koji ima $N + 1$ elemenata (`T[0]` nije definisano) koji predstavlja zahteve korisnika usluga.

vraća `int`: 1 ako svi korisnici usluga mogu biti zadovoljeni, a u suprotnom vraća 0.

Standardni ulaz

Template kod čita ulazne podatke u binarnom formatu. NISTE U MOGUĆNOSTI da menjate ovaj deo koda.

Standardni izlat

Template kod poziva Vašu funkciju više puta sa istim test primerom i to čak i sa različitim železničkim mrežama. Svaki put funkcija mora da vrati tačan odgovor.

Testing

Ovo je tzv. interaktivni problem. Ovaj problem ima mnogo grešaka u formulaciji od početaka. Proverite i poslednju englesku verziju. Ne čitate/pišete po standardnom ulazu/izlazu.

Kod koji obavlja saradnju sa Vašim kodom čita sa standardnog ulaza u sledećem formatu:

- Prva linija sadrži `C`, broj poziva funkcije `solution`.
- Imamo i `C` blokova podataka, koji opisuju parameter funkcije u formatu:

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

- Note that `F[0]`, `F[1]` i `T[0]` se ne pojavljuju na ulazu jer nisu definisani i nemojte ih koristiti tokom rešavanja problema.

- Imaćete na raspolaganju dva ekvivalentna primera: jedan od njih je u text formatu i prati gore opisanu strukturu, a drugi primer je u binarnom formatu i sadrži iste informacije različito kodirane.

Ograničenja i napomene

- Funkcija može da se pozove najviše 50 puta.
- $1 \leq N \leq 1.5 \cdot 10^5$
- $1 \leq S \leq 5 \cdot 10^6$, S = sum of all N within a testcase
- $0 \leq T_i \leq 10^9$ for every $1 \leq i \leq N$

Subtasks

Test cases will be scored **individually**.

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

