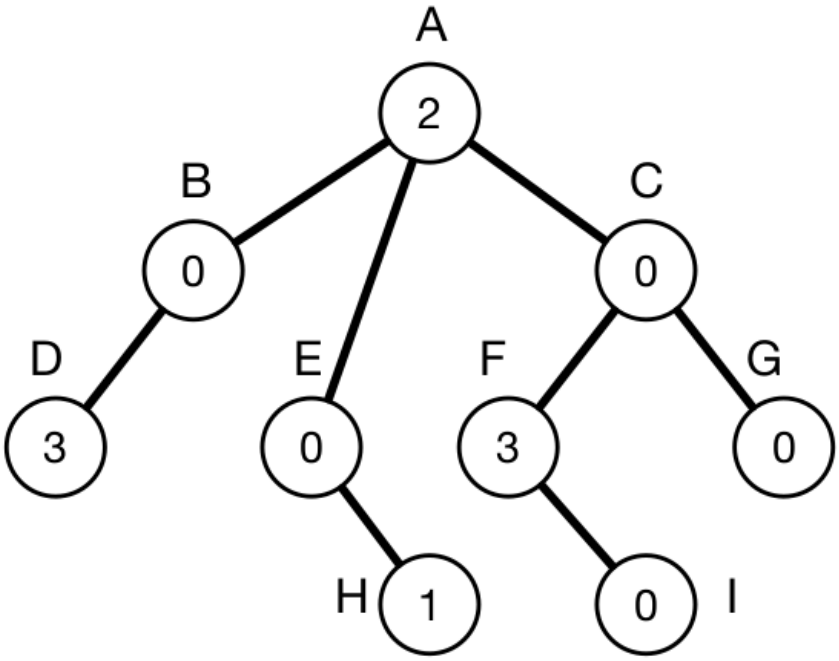
Monete a posto (`monete`)

William colleziona monete, e le tiene in un espositore come quello mostrato in figura, in cui ci sono N contenitori numerati da 1 a N appesi uno all'altro tramite una barra di metallo. In particolare, ogni contenitore tranne quello più in alto (il contenitore numero 1) è appeso a esattamente un altro contenitore. Sua sorella ha giocato con le monete, che prima erano una in ogni posizione, e le ha lasciate nella situazione mostrata qui sotto. William ha però una regola per rimetterle a posto: può spostare una sola moneta per volta, da un contenitore a uno collegato. Quante operazioni sono sufficienti a William per fare in modo che in ogni contenitore ci sia una moneta?



Ad esempio, nella situazione mostrata qui sopra, William impiega:

- Una operazione per spostare una moneta da A a E.
- Una operazione per spostare una moneta da F a I.
- Una operazione per spostare una moneta da F a C.
- Una operazione per spostare una moneta da D a B.
- Quattro operazioni per spostare una moneta da D a G (passando per B, A e infine C).

Dall'esempio qui sopra si vede che servono otto operazioni in totale per rimettere a posto le monete, con una moneta per ogni contenitore.

Dati di input

Il file `input.txt` contiene tre righe. La prima riga contiene l'intero N, il numero di contenitori (e di monete) che ha William. La seconda riga contiene i numeri interi $a_1, \ldots, a_N$, che rappresentano il numero di monete contenute nel rispettivo contenitore dopo che queste sono state disordinate. Infine la terza riga contiene $N-1$ interi $b_2, \ldots, b_N$ che indicano che il contenitore j è appeso al contenitore b_j.

Dati di output

Sul file `output.txt` stampare il numero minimo di spostamenti necessari a William per rimettere a posto tutte le monete.

Assunzioni

- $1 \leq N \leq 100\,000$.
- $0 \leq a_i \leq N$ per ogni $i = 1, \ldots, N$.
- $a_1 + \cdots + a_N = N$.
- $1 \leq b_i < i$ per ogni $i = 1, \ldots, N$.

Esempi di input/output

input.txt	output.txt
9 2 0 0 3 0 3 0 1 0 1 1 2 1 3 3 5 6	8
10 0 0 1 2 1 1 2 0 3 0 1 2 2 2 4 3 5 1 8	11