

Task 2. Miners

Se dă o mină verticală cu N camere, unde fiecare cameră are exact un tunel care intră în ea (dintr-o altă cameră aflată mai aproape de suprafață). Camera cu numărul 1 este conectată direct de suprafață. Formal, mina este un **arbore** cu rădăcina în nodul cu numărul 1.

Fiecare tunel are un scor asociat (care depinde de mulți factori, dar în această problemă ni se dau scorurile direct). Din păcate, uneori trecerea printr-un tunel e riscantă, așa că aceste scoruri pot fi și **negative**.

În momentul de față există un anume număr de mineri în fiecare cameră. Dorim să creăm o alocare minieră, care să includă unii mineri (posibil niciunul), și să alocăm fiecăruia dintre aceștia câte un drum vertical. **Un drum vertical poate merge doar mai adânc** - adică, dacă un miner se află într-o cameră, el poate merge doar în camerele aflate mai departe de suprafață. Formal, dintr-un nod al arborelui putem merge doar în fiii acestuia. Definim scorul unui astfel de drum ca fiind suma scorurilor tunelurilor prin care s-a trecut. Similar, scorul unei alocări este suma scorurilor drumurilor din ea (dacă niciunui miner nu i-a fost alocat un drum, considerăm scorul ca fiind 0).

Totuși, mai avem și alte restricții! Nu dorim să avem camere "aglomerate", adică, pentru fiecare cameră avem o limită pentru numărul maxim de mineri care își pot sfârși drumul în acea cameră. Minerii cărora nu le-a fost alocat niciun drum vor părăsi mina și **nu vor fi considerați pentru aceste restricții**.

Ne interesează cel mai mare scor pe care îl poate avea o alocare minieră. Dându-se structura minei, numărul inițial de mineri din fiecare cameră, și numărul maxim de mineri care-și pot sfârși drumul în fiecare cameră, se cere să scrieți un program numit **miners.cpp**, care să calculeze această valoare.

Date de intrare

De pe prima linie a input-ului standard, programul vostru trebuie să citească un singur întreg N - numărul de camere (considerăm camera 1 ca fiind cea conectată cu exteriorul). A doua linie conține s_1, \dots, s_N - numărul inițial de mineri aflați în fiecare cameră. A treia linie conține e_1, \dots, e_N - numărul maxim de mineri care își pot sfârși drumul în fiecare cameră. În final, ultimele $N - 1$ linii conțin descrierea tunelurilor: a i -a dintre aceste linii conține p_{i+1} și w_{i+1} , semnificând că există un tunel vertical de la camera cu numărul p_{i+1} către camera cu numărul $i + 1$ care are scorul w_{i+1} .

Date de ieșire

Pe o singură linie, programul vostru trebuie să afișeze cel mai mare scor pe care îl poate avea o alocare minieră.

Restricții

$$2 \leq N \leq 5 \times 10^5, 0 \leq s_i, e_i \leq 2000, \text{ pentru orice } 1 \leq i \leq N$$

$$1 \leq p_i < i, \text{ pentru orice } 2 \leq i \leq N$$

$$|w_i| \leq 2000, \text{ pentru orice } 2 \leq i \leq N$$

Subtask-uri

Subtask	Puncte	N	Restricții suplimentare
1	6	≤ 8	Fără restricții suplimentare
2	12	≤ 100	Fără restricții suplimentare
3	14	≤ 2000	Fără restricții suplimentare
4	18	$\leq 10^5$	Arborele formează o linie, adică pentru orice cameră $2 \leq u \leq N$ părintele acesteia este $p_u = u - 1$. De asemenea, $s_u = e_u = 1$ pentru orice $1 \leq u \leq N$.
5	4	$\leq 10^5$	Arborele formează o linie, adică pentru orice cameră $2 \leq u \leq N$ părintele acesteia este $p_u = u - 1$.
6	20	$\leq 10^5$	Fără restricții suplimentare
7	26	$\leq 5 \times 10^5$	Fără restricții suplimentare

Exemplu

Intrare	Ieșire	Explicație
5 5 1 0 0 0 100 1 1 2 4 1 6 1 1 2 2 2 -1	32	O alocare posibilă de scor maxim este: 1) 1 -> 2 -> 4 cu un scor de 8 2) 1 -> 2 -> 4 cu un scor de 8 3) 1 -> 2 cu un scor de 6 4) 1 -> 2 -> 5 cu un scor de 5 5) 1 -> 2 -> 5 cu un scor de 5