

Pushed towards the tight border (pushed_rightw

Visiteremo alcune celle di un array raccogliendo da ciascuna di esse il numero di gemme ivi contenute. Le celle sono numerate da 1 ad n , da sinistra verso destra, e la cella dalla quale partiamo è sempre la cella 1. Per ogni $i = 1, 2, \dots, n$, la cella i contiene g_i gemme, che avremo modo di raccogliere se visiteremo effettivamente tale cella. In ogni cella è presente anche un troll, col troll nella cella i temibile di livello t_i . Quando sono nella cella i faccio appena in tempo a raccogliere le gemme e debbo subito scappare in una cella j con $j > i + t_i$.

Si pianifichi quali celle dell'array visitare per massimizzare il numero di gemme raccolte.

Dati di input

La prima riga del file `input.txt` contiene un numero intero e positivo n , la lunghezza dell'array. La seconda riga contiene una sequenza di n numeri naturali separati da spazio: l' i -esimo di questi numeri è g_i , il numero di gemme nella cella i . La terza riga contiene una sequenza di n numeri naturali separati da spazio: l' i -esimo di questi numeri è t_i , la temibilità del troll della cella i .

Dati di output

Nella prima ed unica riga del file `output.txt` si scriva il massimo numero di gemme che risulta possibile raccogliere.

Esempio di input/output

File <code>input.txt</code>	File <code>output.txt</code>
6 0 1 3 3 0 0 0 2 1 6 0 9	3
File <code>input.txt</code>	File <code>output.txt</code>
6 0 0 2 3 1 1 0 0 1 2 0 5	4

Subtask

- **Subtask 1 [0 punti]:** i due esempi del testo.
- **Subtask 2 [20 punti]:** $n \leq 10$.
- **Subtask 3 [30 punti]:** $t_i = 1$ per ogni cella i , $n \leq 1\,000$.
- **Subtask 5 [20 punti]:** $n \leq 1\,000$.
- **Subtask 6 [30 punti]:** $n \leq 100\,000$.