

Reazione a candela (candele)

Monica ha un elegante set di N candele (del tipo lungo e stretto). Le candele sono indicizzate da 0 a $N - 1$ e, giacché alcune di esse sono parzialmente consumate, hanno lunghezze in generale diverse tra loro. Mojito – il solito distrattone – le ha fatte cadere tutte a terra mentre rincorreva una pallina! Per una fortuita coincidenza, le candele si sono disposte tutte sulla stessa retta (eventualmente sovrapposte). La i -esima candela ha l'estremità superiore della miccia in posizione M_i e la base in posizione B_i , dove M_i e B_i sono numeri interi non negativi.



Inizialmente tutte le candele sono spente. Quando è accesa, una candela brucia al ritmo di una unità al secondo. Quindi, dopo t secondi dall'accensione di una candela la sua miccia si è spostata di esattamente t unità (verso sinistra o verso destra, a seconda dell'orientazione della candela stessa). Nel momento in cui la miccia di una candela accesa si trova nella stessa posizione della miccia di una candela spenta, quest'ultima si accende a sua volta, in una sorta di effetto domino.

Monica decide di accendere la candela di indice 0 al tempo $t_0 = 0$. Aiutala a capire dopo quanti secondi dall'istante iniziale si accende ciascuna candela, o se non si accende affatto.

Implementazione

Dovrai sottoporre un unico file, con estensione `.c` o `.cpp`.

📖 Tra gli allegati a questo task troverai un template `candele.c` e `candele.cpp` con un esempio di implementazione.

Dovrai implementare la seguente funzione:

C	<code>void brucia(int N, int *M, int *B, long long *T);</code>
C++	<code>void brucia(int N, vector<int> &M, vector<int> &B, vector<long long> &T);</code>

- L'intero N rappresenta il numero di candele.
- Il vettore M , indicizzato da 0 a $N - 1$, contiene le posizioni delle micce. Più precisamente, $M[i] = M_i$ per ogni $i = 0, \dots, N - 1$.
- Il vettore B , indicizzato da 0 a $N - 1$, contiene le posizioni delle basi delle candele. Più precisamente, $B[i] = B_i$ per ogni $i = 0, \dots, N - 1$.

La funzione `brucia` dovrà riempire il vettore T , indicizzato da 0 a $N - 1$, con i tempi a cui si accendono le candele. Se una candela non si accende, il suo indice dovrà contenere -1 . **Va specificato anche il tempo a cui si accende la candela 0.**

Grader di prova

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che potete usare per testare le vostre soluzioni in locale. Il grader di esempio legge i dati da `stdin`, chiama le funzioni che dovete implementare e scrive su `stdout`, secondo il seguente formato.

Il file di input è composto $N + 1$ righe, contenenti:

- Riga 1: l'intero N .
- Riga $2 + i$: i due interi M_i e B_i , separati da uno spazio.

Il file di output consiste di una sola riga contenente gli N interi $T[0], T[1], \dots, T[N - 1]$, separati da uno spazio.

Assunzioni

- $1 \leq N \leq 500\,000$.
- $0 \leq M_i, B_i \leq 10^9$ per ogni $i = 0, \dots, N - 1$.
- $M_i \neq B_i$ per ogni $i = 0, \dots, N - 1$.

Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test che lo compongono.

Nel seguito, indicheremo con L il massimo dei $2N$ numeri $M_0, \dots, M_{N-1}, B_0, \dots, B_{N-1}$.

- **Subtask 1 [0 punti]**: Casi d'esempio.
- **Subtask 2 [6 punti]**: Le candele bruciano tutte verso destra (cioè $M_i < B_i$ per ogni i) ed è garantito che tutte le candele prima o poi si accendono (cioè $T[i] \neq -1$ per ogni i).
- **Subtask 3 [10 punti]**: Le candele bruciano tutte verso destra (cioè $M_i < B_i$ per ogni i).
- **Subtask 4 [9 punti]**: $N, L \leq 50$.
- **Subtask 5 [15 punti]**: $N \leq 50\,000, L \leq 200$.
- **Subtask 6 [17 punti]**: $N \leq 3\,000$.
- **Subtask 7 [14 punti]**: Le candele sono lunghe al più 10 (cioè $|M_i - B_i| \leq 10$ per ogni i).
- **Subtask 8 [29 punti]**: Nessuna limitazione specifica.

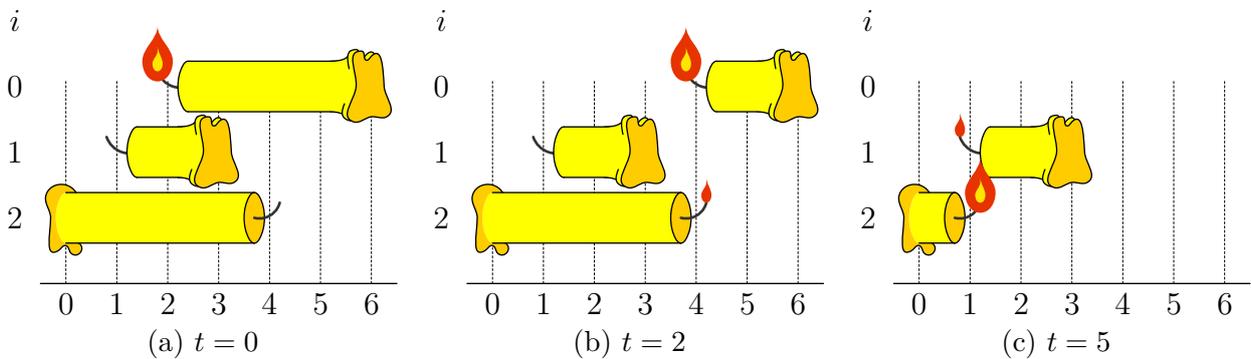
Esempi di input/output

stdin	stdout
3 2 6 1 3 4 0	0 5 2
4 2 5 2 4 2 0 7 2	0 0 0 -1

stdin	stdout
<pre> 8 18 14 5 12 13 22 21 10 8 9 19 12 21 19 15 20 </pre>	<pre> 0 -1 13 21 -1 7 21 3 </pre>
<pre> 4 1 999999998 999999999 1000000000 999999998 0 0 999999999 </pre>	<pre> 0 2999999994 999999997 1999999995 </pre>

Spiegazione

Nel **primo caso di esempio**, la candela 0 inizia a bruciare verso destra al tempo $t = 0$. Al tempo $t = 2$, la sua miccia si trova in posizione $2 + 2 = 4$, che coincide con la posizione della miccia della candela 2. Di conseguenza, quest'ultima candela si accende. Trascorso un altro secondo, la candela 0 si esaurisce, mentre dopo altri 2 secondi (cioè al tempo $t = 5$) la miccia della candela 2 si trova a coincidere con quella della candela 1, la quale è l'ultima ad accendersi.



Nel **secondo caso di esempio**, la miccia della candela 0 coincide con quelle delle candele 1 e 2. Queste ultime, quindi, si accendono immediatamente (cioè al tempo $t = 0$). La candela 3, invece, non viene mai raggiunta dalla miccia di una delle altre candele, e pertanto non si accende mai.

