

Senate Building Decoration (decoration)

La costruzione del nuovo e imponente edificio del Senato è sotto la tua responsabilità: un compito chiave è scegliere la decorazione centrale. Il Senato desidera la decorazione più costosa possibile. Tuttavia, anche i Plebei desiderano partecipare alla scelta e sono preoccupati che una decorazione costosa comporti tasse più alte. Pertanto, desiderano selezionare la decorazione più economica possibile.

Per decidere, viene presentata una fila di N potenziali decorazioni, disposte in linea. Ogni decorazione, indicizzata da 0 a $N - 1$, ha un costo specifico denotato da A_i .

Dopo lunghe discussioni si opta per il seguente metodo decisionale. Il Senato e i Plebei si alternano nella scelta. **Il Senato gioca per primo**. I loro obiettivi sono opposti:

- Il Senato mira a garantire che la decorazione finale abbia il costo **più alto** possibile.
- I Plebei mirano a che la decorazione finale abbia il costo **più basso** possibile.

Ad ogni turno, il giocatore attuale (Senato o Plebei) considera le decorazioni ancora disponibili e seleziona un blocco contiguo (sottoarray) di queste decorazioni. Il blocco selezionato **non può essere l'intero blocco corrente** e deve essere lungo **almeno la metà** (arrotondata per eccesso) della lunghezza del blocco corrente di decorazioni. Tutte le decorazioni che si trovano **al di fuori** del blocco scelto vengono poi rimosse.

Il processo continua con la fila di decorazioni che si restringe dopo ogni turno, finché non rimane una sola decorazione. Il costo di questa decorazione finale è il risultato. Il tuo compito è prevedere questo costo, assumendo sia il Senato che i Plebei giochino in modo ottimale per raggiungere i rispettivi obiettivi.

Implementazione

Dovrai inviare un singolo file sorgente `.cpp`.



Tra gli allegati di questo problema troverai un template `decoration.cpp` con un'implementazione di esempio.

Dovrai implementare la seguente funzione:

C++

```
int decorate(int N, vector<int> A)
```

- L'intero N rappresenta il numero iniziale di potenziali decorazioni.
- L'array A , indicizzato da 0 a $N - 1$, contiene i valori A_0, A_1, \dots, A_{N-1} dove A_i è il costo della i -esima decorazione.
- La funzione deve restituire il costo della decorazione selezionata.

Grader di prova

Tra gli allegati di questo problema troverai una versione semplificata del grader utilizzato durante la valutazione, che puoi usare per testare le tue soluzioni in locale. Il grader di esempio legge i dati da `stdin`, chiama la funzione `decorate` e scrive su `stdout` usando il seguente formato.

Il file di input è formato da 2 righe, contenenti:

- Riga 1: l'intero N .
- Riga 2: N interi separati da spazi: gli elementi A_0, A_1, \dots, A_{N-1} .

L'output è costituito da una singola riga contenente il valore restituito dalla funzione `decorate`.

Assunzioni

- $1 \leq N \leq 5000$.
- $0 \leq A_i \leq 1\,000\,000\,000$.

Assegnazione del punteggio

Il tuo programma verrà testato su diversi casi di test raggruppati in subtask. Per ottenere il punteggio di un subtask, il tuo programma deve risolvere correttamente tutti i suoi casi di test.

- **Subtask 0 [0 punti]**: Casi di esempio.
- **Subtask 1 [12 punti]**: $N \leq 1500$, A è non decrescente (cioè per $i < j$, $A_i \leq A_j$).
- **Subtask 2 [4 punti]**: $N \leq 10$, $A_i \leq 1000$
- **Subtask 3 [22 punti]**: $N \leq 250$
- **Subtask 4 [26 punti]**: $N \leq 800$
- **Subtask 5 [33 punti]**: $N \leq 1500$
- **Subtask 6 [3 punti]**: Nessun vincolo aggiuntivo.

Esempi di input/output

stdin	stdout
5 0 1 2 3 4	3
5 4 3 1 1 5	3

Spiegazione

Nel primo caso di esempio, si può dimostrare che la seguente sequenza di mosse è ottimale:

- Prima mossa del Senato: L'unico sotto-array che massimizza il peggiore esito possibile per loro: $[2, 3, 4]$.
- Risposta dei Plebei: Da $[2, 3, 4]$ devono scegliere un sotto-array di lunghezza almeno due; scegliendo $[2, 3]$ minimizzano il risultato finale.
- Infine il Senato sceglie 3.

Nel secondo esempio:

- Prima mossa del Senato: L'unica apertura che mantiene il loro valore garantito al di sopra di 2 è il blocco $[4, 3, 1]$ (lunghezza 3, almeno la metà di 5 e non l'intera riga).
- Risposta dei Plebei: Da $[4, 3, 1]$ devono mantenere un blocco di lunghezza 2; scegliendo $[3, 1]$ minimizzano il risultato finale.
- Infine il Senato sceglie 3.

Qualsiasi altra prima mossa (ad esempio $[3, 1, 1]$, $[1, 1, 5]$ o un blocco di lunghezza 4) permette ai Plebei di forzare il risultato a 1, quindi il gioco ottimale da entrambe le parti si conclude con costo 3.