

L'uomo che andava appresso ai funerali

A Cocò Alletto un cavò d'ormeggio improvvisamente spezzatosi durante una mareggiata aveva tranciato di netto la gamba mancina. Era omo singolo, che da noi viene a dire tanto magro di corpo quanto privo di pinsèri di moglie e figli e passava le sue giornate su di un paracarro vicino al municipio, sulla via principale di Vigata. La pinsioni che il governo gli passava gli permetteva una dignitosa povirizza.

Un giorno gli passò davanti un funerale sullenne, con la banda in testa e una cinquantina di corone. Non lo seppe mai manco lui perché di colpo venisse pigliato dall'impulso d'accodarsi con il suo passo ballerino. Da quella volta in poi gli divenne un'abitudine. Non fagliava un funerale o una funzione religiosa, acqua cadesse o tirasse vento. Masculi e fimmine, vecchi e picciriddri, funerali e matrimoni, non faceva differenza.

Quando ci fu l'epidemia di influenza maligna che si portò vecchi e picciriddri, a Cocò quasi ci venne l'esaurimento per il travaglio di correre da un capo all'altro della strata, dalla mattina alla sera. Ma ce la fece e non ne perse uno.

Aiuta Cocò Alletto ad individuare la chiesa più vicina nella quale si svolge una determinata funzione. Si supponga che nella via principale di Vigata siano presenti una serie di N chiese, disposte in sequenza, numerate da 0 a $N-1$. In ogni chiesa si svolge un certo tipo di funzione religiosa, identificata da un numero intero non negativo (ad esempio il funerale è identificato dal numero 1, mentre un battesimo è identificato dal numero 3). Cocò si trova in uno delle chiese (ad esempio, la chiesa 7), e si rende conto che deve assistere ad un funerale (ovvero deve recarsi in una delle chiese di tipo 1). Vorrebbe quindi conoscere la distanza minima da percorrere per raggiungere una chiesa di tipo 1 partendo dalla chiesa 7. Si supponga che la distanza tra due chiese adiacenti sia sempre pari a 1. Ad esempio, nel caso in cui la chiesa 7 sia già una chiesa di tipo 1, la distanza restituita sarà 0.

Il programma dovrà rispondere a Q domande, ognuna delle quali fornirà una coppia di valori (a, b) , dove a è la posizione corrente (cioè il numero della chiesa in cui si trova Cocò) e b è il tipo di chiesa in cui si svolge la prossima funzione. La risposta del programma sarà la distanza tra la posizione a e la posizione della chiesa di tipo b più vicino alla chiesa a .

Nota: le domande sono indipendenti, l'effetto di una domanda non influisce sulle successive.

Dati di input

Il file `input.txt` è composto da $Q+2$ righe. La prima riga contiene due interi separati da spazio: N, Q . La seconda riga contiene gli N interi t_i separati da uno spazio. Seguono Q righe che contengono due interi ciascuna: i parametri a_i e b_i dell' i -esima domanda.

Dati di output

Il file `output.txt` è composto da Q righe. La i -esima riga contiene la risposta alla i -esima query.

Assunzioni

- $1 \leq N, Q \leq 100.000$.
- $0 \leq t_i < 100.000$.
- $0 \leq a_i \leq N-1$.
- È sempre garantito che esiste almeno una chiesa di tipo b_i .

Limitazioni

Limiti di tempo : 1 secondo

Limiti di spazio: 256 MB

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 relativi ad esso.

- **Subtask 1 [10 punti]:** Casi d'esempio.
- **Subtask 2 [40 punti]:** $N, Q \leq 5000$.
- **Subtask 3 [30 punti]:** $N \leq 5000$.
- **Subtask 4 [20 punti]:** Nessuna limitazione specifica.

Esempio di input/output

input.txt	output.txt
6 3 0 1 2 1 2 0 2 0 3 0 1 2	2 2 1
10 6 23 5 29 3 3 5 2 7 11 13 0 5 3 5 5 5 9 5 0 3 9 3	1 2 0 4 3 5
2 3 10 20 0 10 0 20 1 10	0 1 1