

Esame algoritmi 2019-02-26 VR

Da n a zero ...

in meno mosse possibili (bit_edit_to_zero)

Questo problema è un rimastering di un problema di HackerRank. Ricevi in input un numero n e devi trasformarlo nel numero 0 impiegando il minor numero possibile di mosse. Hai a disposizione due possibili mosse, entrambe meglio descritte guardando alla rappresentazione in binario del numero n :

Mossa 1: inverte il valore del bit più a destra, ossia il bit di parità (quello meno significativo);

Mossa 2: inverte il valore del bit alla immediata sinistra del bit posto più a destra tra quelli settati ad uno. (Subito a destra del bit che si modifica deve esserci un bit settato ad 1, e non vi è alcun bit settato ad uno a destra di questo).

Devi implementare le seguenti due funzioni:

Funzione 1 `num_mosse(n)` deve ritornare il minimo numero di mosse per passare da n a zero;

Funzione 2 `mossa(n)` ritorna 1 oppure 2: la risposta i ($=1,2$) è corretta se la mossa i , applicata ad n , ci porta un passo più vicini allo zero.

Dati di input

Il vostro programma, nell'ordine, riceve in input:

- un numero naturale $p \in \{1,2\}$, che specifica la funzione da chiamare
- il numero naturale n da cui prende avvio il processo

Dati di output

- se $p = 1$ il programma restituisce in output il numero minimo di mosse da effettuare per ridurre a 0 il valore di partenza n ricevuto in input, ossia il valore restituito dalla funzione `num_mosse(n)`
- se $p = 2$ il programma restituisce in output il valore restituito dalla funzione `mossa(n)`, ossia un valore $i \in \{1,2\}$ che risulta corretto se la mossa i ci porta un passo più vicini allo zero

Esempio di input/output

In attachment alla pagina del problema trovate diverse copie input/output tra cui le seguenti.

File input.txt	File output.txt
1 2	3
File input.txt	File output.txt
2 2	1

File input.txt	File output.txt
2 3	2
File input.txt	File output.txt
1 5	6
File input.txt	File output.txt
2 5	2

Assunzioni e note

NOTA: abbiamo sopra fornito una descrizione del formato di input/output soltanto per facilitarvi il test sul vostro computer. Per sottomettere il problema è obbligatorio utilizzare il template che potete scaricare fra gli allegati del problema, avendo cura di modificare solamente l'implementazione delle funzioni richieste. Questo è necessario per garantire la compatibilità del vostro programma con il sistema di valutazione (se ad esempio utilizzate python potrebbe utilizzare una versione di python diversa, quale python2).

Subtask

- **Subtask 1 [0 punti]:** gli esempi del testo.
- **Subtask 2 [10 punti]:** $p = 1$, $n = 2^k$, $k \leq 10$.
- **Subtask 3 [20 punti]:** $p = 1$, $n \leq 1000$.
- **Subtask 4 [20 punti]:** $p = 1$, $n \leq 100\,000$.
- **Subtask 5 [10 punti]:** $p = 2$, $n = 2^k$, $k \leq 10$.
- **Subtask 6 [20 punti]:** $p = 2$, $n \leq 1000$.
- **Subtask 7 [10 punti]:** $p = 2$, $n \leq 1\,000\,000$.
- **Subtask 8 [10 punti]:** $p = 2$, $n \leq 100\,000\,000$.