

oddKnap

Descrizione del problema

Data una sequenza di n numeri naturali p_1, \dots, p_n ed un budget $B \in \mathbb{N}$, viene chiesto di trovare una sottosequenza composta da un numero dispari di questi e di somma non eccedente B ma altrimenti massima. In realtà i numeri p_1, \dots, p_n sono i pesi di n oggetti indicizzati da 1 ad n . Le sottosequenze sono quindi in corrispondenza biunivoca con i sottoinsiemi dell'insieme di questi oggetti e su di esse possiamo quindi ereditare l'ordinamento lessicografico che vede $S_1 < S_2$ se l'oggetto di indice minimo nella differenza simmetrica $S_1 \Delta S_2$ appartiene ad S_1 . Per fare punteggio pieno, oltre a restituire il valore massimo per la somma, il numero di soluzioni ottime modulo 10^5 (ossia le ultime 5 cifre del numero di soluzioni ottime).

Dati di input

Sulla prima riga, il file input.txt contiene n (= numero di oggetti) e B (= budget), separati da uno spazio. Sulla seconda riga, i numeri p_1, \dots, p_n separati da uno spazio.

Dati di output

Se la prima riga del file output.txt contiene la massima somma, non eccedente B , di una sottosequenza di p_1, \dots, p_n di lunghezza dispari, per quell'istanza ti sei già aggiudicato un punteggio parziale che ammonta ad 1/2.

Se nella seconda riga del file output.txt riporti il numero di soluzioni ottime modulo 100.000 (ossia le ultime 5 cifre del numero di soluzioni ottime), allora ti sei aggiudicato punteggio pieno su quell'istanza.

Assunzioni

- limite all'user time: 1 secondo;
- $1 \leq N \leq 1000$;
- $B \leq 10000$;
- $1 \leq p_i \leq 1000$ per il peso p_i di ciascun oggetto;
- la prima riga dell'output, se corretta, garantisce l'ottenimento di un **punteggio parziale** (almeno mezzo punto per istanza).
- se il programma utilizza più memoria di quanta ne serva per allocare 10^6 interi, si applicherà un punteggio parziale ridotto.

Esempi di input/output

File input.txt	File output.txt
12 21 7 2 5 6 4 3 5 9 1 8 6 3	21 60
File input.txt	File output.txt
6 17 7 7 5 3 7 8	17 3