

Numeri e triangoli (triangoli)

Limite di tempo: 2.0 secondi

Limite di memoria: 256 MiB

Luca sta ripassando matematica in vista dell'esame di maturità con un suo compagno di classe, Matteo. Dopo un po' di tempo però, presi dalla noia, cominciano a scarabocchiare qualche numero e qualche figura geometrica a caso e si inventano al momento un gioco interessante.

Il gioco è stato così deciso dai due amici: Matteo scarabocchia N numeri sul foglio e Luca deve cercare di sceglierne tre in modo che **NON** possano rappresentare le lunghezze dei tre lati di un triangolo valido. Se Luca riesce a trovare anche solo una combinazione ha vinto, altrimenti ha vinto Matteo.

☞ Tre numeri sono lunghezze valide per i lati di un triangolo se ciascuno è minore o uguale alla somma degli altri due.

☞ In questa situazione viene considerato triangolo anche un cosiddetto “triangolo degenere” (ovvero quando un lato misura quanto la somma degli altri due).

Sapresti contare quante possibili combinazioni Luca potrebbe trovare?

Dati di input

Il file `input.txt` è composto da due righe. La prima riga contiene l'intero N . La seconda riga contiene N interi separati da uno spazio, i numeri V_i che Matteo ha scritto sul foglio.

Dati di output

Il file `output.txt` è composto da un'unica riga contenente un unico intero, la risposta a questo problema.

Assunzioni

- $3 \leq N \leq 15\,000$.
- $1 \leq V_i \leq 10\,000\,000$ 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 relativi ad esso.

- **Subtask 1 [10 punti]:** Casi d'esempio.
- **Subtask 2 [30 punti]:** $N \leq 1000$.
- **Subtask 3 [40 punti]:** $N \leq 5000$.
- **Subtask 4 [20 punti]:** Nessuna limitazione specifica.

Esempi di input/output

input.txt	output.txt
3 1 2 3	0
input.txt	output.txt
4 3 4 10 6	2

Spiegazione

Nel **secondo caso di esempio**, le combinazioni con cui NON si ottiene un triangolo sono:

- 3 - 4 - 10
- 3 - 6 - 10

Teorema dei quattro quadrati (qquadrati)

Limite di tempo: 1.0 secondi

Limite di memoria: 256 MiB

Luca sta ripassando matematica in vista dell'esame di maturità e in particolare sta studiando il teorema di Lagrange relativo alle funzioni continue e derivabili. Incuriosito dal lavoro del matematico italiano è andato a cercare quali altri teoremi gli vengono attribuiti e ne ha trovato uno interessante.

Il teorema in questione è detto “Teorema dei quattro quadrati” e afferma che ogni intero positivo può essere espresso come somma di (al più) quattro quadrati perfetti. Formalmente:

$$n = a^2 + b^2 + c^2 + d^2 \quad \forall n \in \mathbb{N}$$

dove a, b, c, d sono interi non negativi.

Luca però non riesce a trovare un modo per sapere in quanti modi diversi n può essere ottenuto seguendo quanto afferma il teorema. Aiuta Luca!

Dati di input

Il file `input.txt` è composto da due righe. La prima riga contiene l'intero N . La seconda riga contiene N interi separati da uno spazio, i numeri V_i su cui applicare il teorema.

Dati di output

Il file `output.txt` è composto da un'unica riga contenente N interi: l' i -esimo numero indica in quanti modi diversi si può ottenere V_i seguendo il teorema.

Assunzioni

- $1 \leq N \leq 200$.
- $1 \leq V_i \leq 2^{15}$ 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 relativi ad esso.

- **Subtask 1 [10 punti]:** Casi d'esempio.
- **Subtask 2 [40 punti]:** $N \leq 30$.
- **Subtask 3 [50 punti]:** Nessuna limitazione specifica.

Esempi di input/output

input.txt	output.txt
1 25	3
input.txt	output.txt
2 10 31	2 2

Spiegazione

Nel **primo caso di esempio**, 25 si può ottenere con le tre seguenti combinazioni:

- $25 = 1^2 + 2^2 + 2^2 + 4^2$
- $25 = 3^2 + 4^2 (+0^2 + 0^2)$
- $25 = 5^2 (+0^2 + 0^2 + 0^2)$

Massima latenza (latenza)

Limite di tempo:	1.0 secondi
Limite di memoria:	256 MiB

La classe di Luca sta escogitando un modo per comunicare di nascosto durante le prove scritte dell'esame di maturità. I rigorosi commissari esterni hanno spento la rete Wi-Fi dell'istituto e controllano attentamente che nessuno parli. La commissione ha però commesso un grave errore: non ha ritirato gli smartphone.

Per riuscire comunque a scambiarsi informazioni segretamente, la classe ha intenzione di sfruttare la comunicazione via Bluetooth tra un telefono e l'altro. Come è ben noto, la comunicazione via Bluetooth richiede che i due dispositivi vengano prima accoppiati. Poiché il tempo prima dell'inizio della prova scritta è limitato ma si vuole che ciascuno studente sia connesso al resto della classe, la classe esegue il *numero minimo* di accoppiamenti tra telefoni sufficienti affinché *nessuno* resti escluso.

Ovviamente nella classe ci sono diversi smartphone, funzionanti a velocità differenti. Questo comporta che i collegamenti via Bluetooth (per loro natura *bidirezionali*) abbiano velocità e latenze diverse. La latenza è il tempo che il segnale impiega per passare dal cellulare i al cellulare j .

Come è naturale che sia, se due smartphone i e j non sono direttamente accoppiati (ma c'è tra loro k) definiamo la latenza totale come la somma delle latenze. Quindi $L(i, j) = L(i, k) + L(k, j)$.

Luca però intravede un possibile problema: due smartphone potrebbero essere molto distanti e quindi potrebbero esserci significativi ritardi nella comunicazione. Ciò comprometterebbe irrimediabilmente il buon esito della prova d'esame. Aiuta Luca a calcolare *la massima latenza* (in millisecondi) tra due smartphone.

Dati di input

Il file `input.txt` è composto da N righe. La prima riga contiene l'intero N , il numero di smartphone. Le successive $N - 1$ righe descrivono ciascuna l' i -esimo accoppiamento. In particolare ciascuna riga contiene tre interi separati da spazio che indicano l'esistenza di una connessione tra A_i e B_i con una latenza L_i (in millisecondi).

Dati di output

Il file `output.txt` contiene un singolo intero: la *massima latenza* (in millisecondi) tra due smartphone.

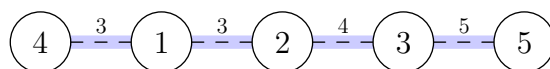
Assunzioni

- $2 \leq N \leq 100\,000$.
- $1 \leq A_i, B_i \leq N$ per ogni $i = 0 \dots N - 1$.
- $1 \leq L_i \leq 1\,000$ 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 relativi ad esso.

- **Subtask 1 [10 punti]:** Casi d'esempio.
- **Subtask 2 [15 punti]:** Gli smartphone sono collegati in sequenza, come nella figura seguente (che rappresenta il secondo il caso di esempio)



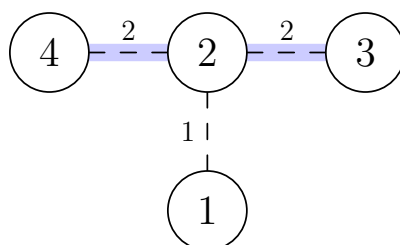
- **Subtask 3 [30 punti]:** $N \leq 200$.
- **Subtask 4 [25 punti]:** $N \leq 2500$.
- **Subtask 5 [20 punti]:** Nessuna limitazione specifica.

Esempi di input/output

input.txt	output.txt
4 2 3 2 4 2 2 2 1 1	4
input.txt	output.txt
5 2 3 4 4 1 3 3 5 5 1 2 3	15

Spiegazione

L'immagine rappresenta il primo caso di esempio.



La massima latenza si ha tra gli smartphone 1 e 3 e vale 4 millisecondi (il percorso è evidenziato in blu). Tutte le altre coppie sono separate da una latenza inferiore.

Interrogazioni ordinate (interrogazioni)

Limite di tempo: 1.0 secondi
Limite di memoria: 256 MiB

La commissione istituita per l'esame di maturità della classe di Luca ha ricevuto dal MIUR l'elenco dei codici SIDI di tutti gli N alunni da esaminare, con il suggerimento che l'ordine delle interrogazioni all'esame orale segua, in ordine crescente, il codice SIDI. Ciò significa che uno studente i dovrebbe essere interrogato prima di uno studente j se il codice SIDI del primo è inferiore a quello del secondo (ovvero $C_i < C_j$).

Seguire le indicazioni del Ministero comporterebbe però una certa confusione tra gli studenti, abituati a essere interrogati in ordine per numero crescente di registro (numero 1, numero 2, ..., numero N).

Assecondando le richieste degli studenti, la commissione stabilisce di adottare la modalità consueta (in ordine per numero crescente di registro), ma vuole prima stabilire qual è l'impatto di questa scelta. In particolare si vuole stabilire per quante coppie di studenti (i, j) cambierebbe l'ordine di interrogazione, ovvero per quante coppie (i, j) i verrebbe interrogato prima di j con uno dei due metodi ma dopo j con l'altro.

Dati di input

Il file `input.txt` è composto da 2 righe. La prima riga contiene l'intero N , il numero di studenti. La seconda riga contiene N interi positivi separati da spazio: i codici SIDI degli alunni dal numero 1 al numero N .

Dati di output

Il file `output.txt` contiene un singolo intero: il **numero di coppie** di studenti che cambiano l'ordine relativo di interrogazione tra il primo e il secondo metodo.

☞ Attenzione: la risposta al problema supera, in alcuni casi, 2^{32} . Se si lavora, ad esempio, in C/C++ è quindi richiesto l'uso del tipo `long long` al posto di `int`.

Assunzioni

- $2 \leq N \leq 200\,000$.
- $1 \leq C_i \leq 10\,000\,000$ per ogni $i = 0 \dots N - 1$.
- Il codice SIDI è univoco: $C_i \neq C_j$ per ogni (i, j) con $i \neq j$.

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]:** $N \leq 1000$
- **Subtask 3 [25 punti]:** Gli N studenti hanno un codice SIDI compreso tra 1 e N (ovvero $1 \leq C_i \leq N$ per ogni $i = 0 \dots N - 1$).
- **Subtask 4 [25 punti]:** Nessuna limitazione specifica.

Esempi di input/output

input.txt	output.txt
4 14 63 22 31	2
input.txt	output.txt
5 84 31 57 25 66	6

Spiegazione

Nel primo esempio considera le seguenti coppie di studenti il cui ordine cambia tra i due metodi:

- Codici SIDI 63 e 22 ($63 > 22$), sono rispettivamente i numeri 2 e 3 nel registro ($2 < 3$).
- Codici SIDI 63 e 31 ($63 > 31$), sono rispettivamente i numeri 2 e 4 nel registro ($2 < 4$).

Tutte le altre coppie mantengono la stessa posizione relativa: gli studenti della coppia verranno interrogati nello stesso ordine indipendentemente dal metodo utilizzato.