

Corso di Laurea Magistrale in Ingegneria Informatica

CORSO DI
ALGORITMI E STRUTTURE DATI
Prof. ROBERTO PIETRANTUONO

Indicazioni

Si consegna un file in **formato editabile (.txt, .docx, .rtf, etc.)** nominandolo "*CognomeNome*", in cui è riportata l'implementazione (nel linguaggio scelto) seguita da una indicazione della complessità temporale dell'algoritmo implementato (complessità nel caso peggiore, è sufficiente il limite superiore $O(f(n))$). Se si utilizzano librerie di cui non si conosce la complessità, lo si indichi nella spiegazione (ad esempio, "la complessità è $O(n \log n)$ al netto della complessità dell'algoritmo x , che è non nota"). Se si utilizza la randomizzazione, si indichi anche il tempo di esecuzione atteso.

PROBLEMA

Una progressione aritmetica è una successione di numeri tali che la differenza tra ciascun termine (o elemento) della successione e il suo precedente è costante. La somma degli elementi in una progressione di n elementi è data da: $S = \frac{1}{2} * n * (a_1 + a_n)$, dove a_1 ed a_n sono il primo e l'ultimo elemento. E' dato in input un array di interi positivi che rappresenta una progressione aritmetica dove però manca un elemento. La massima lunghezza è 30, il massimo valore è 1000. Si implementi un algoritmo che trova l'elemento mancante.

Suggerimento: una soluzione semplice si ottiene scorrendo il vettore, complessità $O(N)$. Si utilizzi un algoritmo *divide et impera* con complessità $O(\log N)$.

INPUT

La prima riga contiene il numero di casi di test N . Ogni caso di test, che inizia alla riga successiva, è composto da due righe: la prima riportante la dimensione del vettore, la seconda riportante gli elementi del vettore separati da uno spazio.

OUTPUT

Si stampi, per ogni caso di test, il numero mancante.

Sample Input

```
2
4
1 3 5 9
6
3 6 12 15 18 21
```

Sample Output

```
7
9
```