

Esercizi Complessità

1. Si scriva un algoritmo che dato in input un array di n interi e un intero k , determini se esiste almeno una coppia fra gli elementi la cui somma è esattamente k . Si stimi poi la complessità della procedura proposta.
2. Elencare in ordine crescente di complessità asintotica le seguenti funzioni:

- n
- $n \log n$
- $2^{\log \log n}$
- \sqrt{n}
- $2^{3+\log n}$

3. Per uno stesso problema sono stati disegnati tre diversi algoritmi le cui complessità in tempo sono:

- $f_1(n) = \sum_{k=1}^n (k^2 + k)$
- $f_2(n) = \sum_{k=0}^{\lfloor \log_2 n \rfloor} \left\lceil \frac{n^2}{2^k} \right\rceil$
- $f_3(n) = \begin{cases} T(n-2) + 4 & n \geq 2 \\ 1 & n = 1 \end{cases}$

Qual'è l'algoritmo più efficiente? Motivare adeguatamente la risposta.

4. Risolvere la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} T(n-2) + O(\sqrt{n}), & n > 5 \\ O(1), & n \leq 5 \end{cases}$$