

Algoritmi di ricerca sequenziale

- Permettono di trovare un elemento in una struttura dati
- Controllano ogni elemento uno per uno
- Metodo semplice e intuitivo
- Adatto a piccoli insiemi di dati

Principio di funzionamento

- Scorrimento dell'elenco dall'inizio alla fine
- Confronto di ogni elemento con il valore cercato
- Se trovato, restituisce posizione
- Se non trovato, segnala assenza

Esempio passo-passo

- Lista: [5, 3, 8, 2]
- Ricerca elemento 8
- Controlla: $5 \neq 8 \rightarrow 3 \neq 8 \rightarrow 8 = 8 \rightarrow$ trovato
- Restituisce indice = 2

Pseudocodice

```
per i da 0 a n-1:  
    se lista[i] == elemento_cercato:  
        ritorna i  
ritorna -1 # elemento non trovato
```

- Mostra semplicità dell'algoritmo
- Facile da implementare in qualsiasi linguaggio

Complessità

- Tempo: $O(n)$
- Memoria: $O(1)$
- Dipende dalla posizione dell'elemento
- Efficienza bassa su grandi dataset

Vantaggi

- Implementazione semplice
- Non richiede dati ordinati
- Utile per didattica e problemi base
- Metodo universale

Svantaggi

- Inefficiente su dataset grandi
- Numero di confronti elevato
- Migliorabile con ricerca binaria su dati ordinati
- Limitato a contesti semplici