

Algoritmi di ordinamento per selezione

- Ordinano una sequenza di elementi
- Basati su confronto e scambio
- Semplici da implementare
- Adatti a piccoli insiemi di dati

Principio di funzionamento

- Scorre l'elenco per trovare il minimo (o massimo)
- Lo scambia con il primo elemento non ordinato
- Ripete il processo per tutti gli elementi
- Metodo iterativo e intuitivo

Esempio passo-passo

- Lista iniziale: `[5, 3, 8, 2]`
- Primo ciclo: minimo = 2 → scambiato con 5
- Lista: `[2, 3, 8, 5]`
- Ripetere per posizione successiva

Pseudocodice

```
per i da 0 a n-1:  
    min_idx = i  
    per j da i+1 a n:  
        se lista[j] < lista[min_idx]:  
            min_idx = j  
    scambia lista[i] con lista[min_idx]
```

- Evidenzia il concetto di minimo e scambio
- Base per comprendere altri algoritmi

Complessità

- Tempo: $O(n^2)$
- Memoria: $O(1)$
- Poco efficiente su grandi dataset
- Adatto a didattica e piccoli esempi

Vantaggi

- Semplice da capire e implementare
- Non richiede memoria aggiuntiva
- Buono per insegnamento e algoritmi introduttivi
- Metodo deterministico

Svantaggi

- Inefficiente su grandi quantità di dati
- Confronti e scambi elevati
- Non adatto a contesti professionali complessi
- Meglio algoritmi più avanzati per grandi dataset