
1. Bubble Sort (Ordinamento a Bolla)

Cosa fa?

Confronta coppie di elementi adiacenti e li scambia se sono nell'ordine sbagliato.

Di solito lo fa partendo "dal più piccolo" e ordinando in base a quello.

Ripete questo processo finché l'array non è ordinato.

Schema buono: https://www.w3schools.com/dsa/dsa_algo_bubblesort.php

Come funziona?

- **Esempio con numeri:**

Array da ordinare: [5, 3, 8, 4]

- Prima passata: 3 e 5 → già in ordine. Poi 8 e 4 → scambio → [5, 3, 4, 8].
- Seconda passata: 3 e 5 → scambio → [3, 5, 4, 8].
- Terza passata: 5 e 4 → scambio → [3, 4, 5, 8].
- Finito!

Codice Java (semplificato):

```
void bubbleSort(int[] arr) {  
    for (int i = 0; i < arr.length - 1; i++) {  
        for (int j = 0; j < arr.length - i - 1; j++) {  
            if (arr[j] > arr[j + 1]) { // Se il numero è più grande del  
successivo...  
                // Scambia i due numeri  
                int temp = arr[j];  
                arr[j] = arr[j + 1];  
                arr[j + 1] = temp;  
            }  
        }  
    }  
}
```

Schema Visivo (Testuale):

Prima passata: [5, 3, 8, 4] → [5, 3, 4, 8]

Seconda passata: [3, 5, 4, 8]

Terza passata: [3, 4, 5, 8]

Quando usarlo?

- Per piccoli dataset o esempi didattici.
- **Velocità:** Lento per grandi dati (complessità $O(n^2)$).

2. Insertion Sort (Ordinamento per Inserimento)

Cosa fa?

Prende un elemento alla volta e lo "inserisce" nella posizione corretta all'interno della parte già ordinata dell'array.

Link buono: <https://www.geeksforgeeks.org/insertion-sort-algorithm/>

Come funziona?

- **Esempio con numeri:**

Array da ordinare: [5, 3, 8, 4]

- Parte dal secondo elemento (3). Lo confronta con 5 → sposta 5 a destra e inserisci 3 → [3, 5, 8, 4].
- Prendi 8: già in ordine → [3, 5, 8, 4].
- Prendi 4: confronta con 8, 5, 3 → sposta 8 e 5 a destra, inserisci 4 → [3, 4, 5, 8].

Codice Java (semplificato):

```
void insertionSort(int[] arr) {
    for (int i = 1; i < arr.length; i++) {
        int key = arr[i]; // Elemento da inserire
        int j = i - 1;    // Indice della parte già ordinata
        // Sposta gli elementi maggiori di "key" verso destra
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key; // Inserisci "key" nella posizione corretta
    }
}
```

Schema Visivo (Testuale):

Step 1: [5, 3, 8, 4] → [3, 5, 8, 4]

Step 2: [3, 5, 8, 4] → [3, 4, 5, 8]

Step 3: [3, 5, 8, 4] → [3, 4, 5, 8]

Quando usarlo?

- Quando l'array è quasi ordinato.
- Più veloce di Bubble Sort su dati piccoli.
- **Velocità:** Lento per grandi dati (complessità $O(n^2)$).

Confronto tra i Due Algoritmi

Caratteristica	Bubble Sort	Insertion Sort
Logica	Confronta e scambia vicini	Sposta elementi per inserire
Migliore Caso	$O(n)$ (su array già ordinato)	$O(n)$ (su array quasi ordinato)
Peggior Caso	$O(n^2)$	$O(n^2)$
Usabilità	Didattico	Piccoli dataset o quasi ordinati

Esempi Pratici

Bubble Sort per Ordinare Un Elenco di Nomi:

```
String[] nomi = {"Marco", "Alessia", "Luca"};  
// Applica Bubble Sort per metterli in ordine alfabetico
```

Insertion Sort per Ordinare Carte in Una Mano:

Immagina di tenere in mano delle carte disordinate. Prendi una carta alla volta e la infili nel posto giusto tra quelle già ordinate.

Ricorda!

- **Bubble Sort** è come "gareggiare a scambiarsi di posto".
- **Insertion Sort** è come "aggiustare un mazzo di carte mano a mano che le ricevi".
- Entrambi hanno complessità $O(n^2)$, quindi non usateli per grandi dataset (usate algoritmi più veloci come Quick Sort o Merge Sort).

Se hai dubbi, chiedi pure! 😊