



UNIVERSITÀ  
DI TRENTO

Dipartimento di Ingegneria e  
Scienza dell'Informazione  
DISI - Trento

# Programmazione 1

## 13 - Esercitazione

Giovanna Varni  
[giovanna.varni@unitn.it](mailto:giovanna.varni@unitn.it)

Anno Accademico 2023/2024

# 00 - Searching

Un problema frequente è la ricerca di un elemento in un array di  $N$  elementi, per questo è importante utilizzare tecniche di ricerca affidabili ed efficienti.

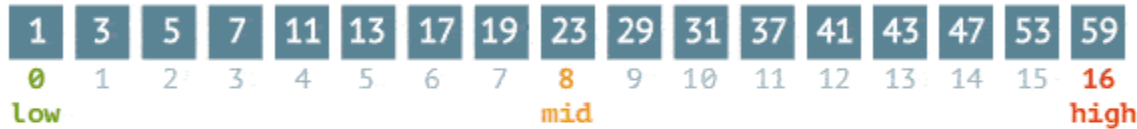
Se l'array **non** è ordinato  $\Rightarrow$  ricerca sequenziale ( $O(n)$ )

Se l'array è ordinato  $\Rightarrow$  ricerca binaria ( $O(\log(n))$ )

# 00 - Searching

Binary search

steps: 0



Sequential search

steps: 0



[www.penjee.com](http://www.penjee.com)

# 00 - Sorting

## Algoritmi di Ordinamento:

- Bubble Sort
- Merge Sort
- Quicksort
- Heapsort
- Selection Sort
- Shellsort
- Tree Sort
- Bucket Sort
- Pigeonhole Sort
- Spaghetti Sort
- Random Sort
- ...

# 00 - Sorting

Avere array ordinati garantisce efficienza in:

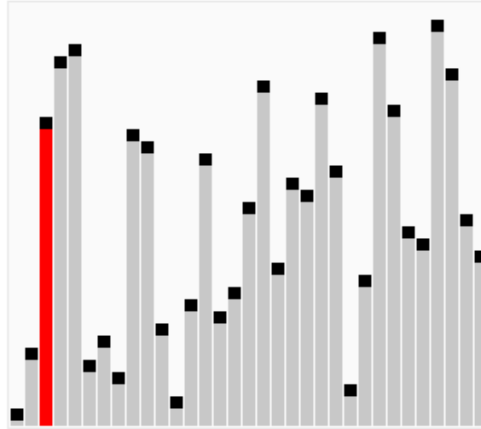
- Ricerca di un elemento in un array
- Merge di due array
- Inserimento ordinato
- ...

# 01 - Ancora su array...

**Scrivere un programma che, chiesto in input all'utente un numero intero e una base che può assumere valore 2 oppure 8, stampi a video il numero e lo converta nella notazione binaria o ottale corrispondente ad 8 cifre. Nel caso il numero sia negativo o troppo grande per essere rappresentato a 8 cifre, il programma darà un messaggio di errore e chiederà all'utente di inserire un nuovo numero. Il programma stampi a video la codifica binaria o ottale del numero.**

## 02 - Bubble Sort

**Scrivere un programma che dichiari e poi inizializzi un array di 10 interi con valori randomici da 0 a 99 inclusi e lo stampi a video. Ordinare poi l'array implementando l'algoritmo Bubble Sort avendo cura di eliminare iterazioni inutili qualora l'array risulti ordinato. Si stampi a video, inoltre, il numero di iterazioni necessarie a ordinare l'array.**



**Se ne implementi in seguito la versione ricorsiva, stampando a video il numero di passi di ricorsione effettuati.**

## 03 - Duplicati

**Scrivere un programma che, dato un array di 10 numeri interi (da 0 a 9), lo ordini e poi sostituisca eventuali numeri duplicati con il valore -1.**

`[3, 0, 4, 0, 3, 7, 5] ==> [0, -1, 3, -1, 4, 5, 7]`



# 04 - Merge

**Scrivere un programma che, dati due array di dimensioni diverse, li ordini e ne faccia il merge in un terzo array di dimensioni pari alla somma delle dimensioni dei due array in input.**

# 05 - Binary Search

**Scrivere un programma che, dato un array di 10 interi (da 0 a 9) ordinato, chieda in input all'utente un intero da 0 a 9 ed implementi la ricerca binaria tramite funzione ricorsiva per trovare la posizione (indice) della prima occorrenza del numero in input.**