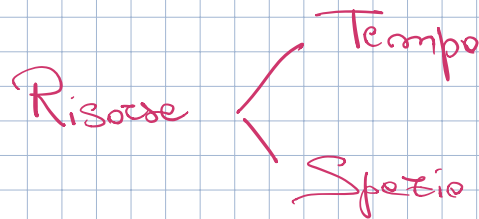


Argomenti

- Strutture Dati Statiche
- Problemi di Ordinamento - Selezione
- Strutture Dati Dinamiche
Liste, Pile, Code, Code con Priorità
Tabelle di hash, Alberi di Ricerca
- Studio di Complessità
- Correttezza
- Grafi, Algoritmi di Visite, Cammini Minimi ...



Problema → Algoritmo

Modello calcolo

Pseudo-codice

ORDINAMENTO

Dati n n.ri interi $\langle a_1, a_2, \dots, a_n \rangle$
determinare una permutazione π degli
indici $\{1, \dots, n\}$ tale che

$$a_{\pi(1)} \leq a_{\pi(2)} \leq \dots \leq a_{\pi(n)}$$

\Uparrow

Esempio

$$\begin{matrix} 1 & 2 & 3 & 4 \\ \langle 5, & 8, & 2, & 1 \rangle \end{matrix}$$

$$\langle 1, 2, 5, 8 \rangle$$

$$\pi(1) = 4$$

$$\pi(2) = 3$$

$$\pi(3) = 1$$

$$\pi(4) = 2$$



Gli n elementi sono memorizzati in un

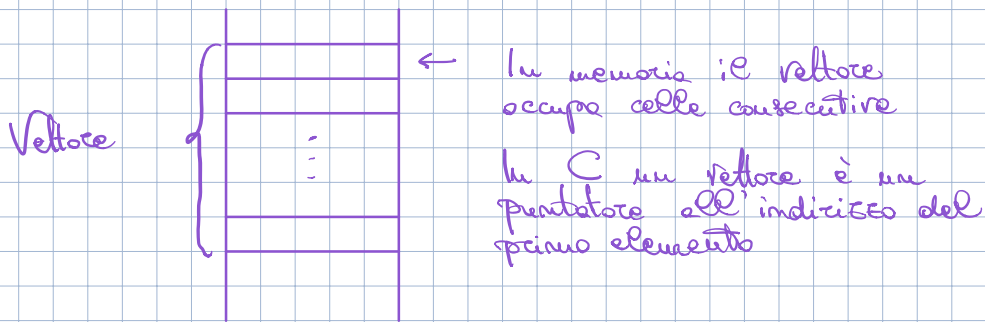
VETTORE (ARRAY)

Struttura dati statica omogenea

lineare ad accesso diretto

$\{$

ho cambiato da sequenziale a lineare
in modo che non ci sia confusione
con il tipo di accesso



Ordinamento

Dato un vettore A di lunghezza n , permutare gli elementi di A in modo che si termini

$$A[1] \leq A[2] \leq \dots \leq A[n]$$

Selection Sort

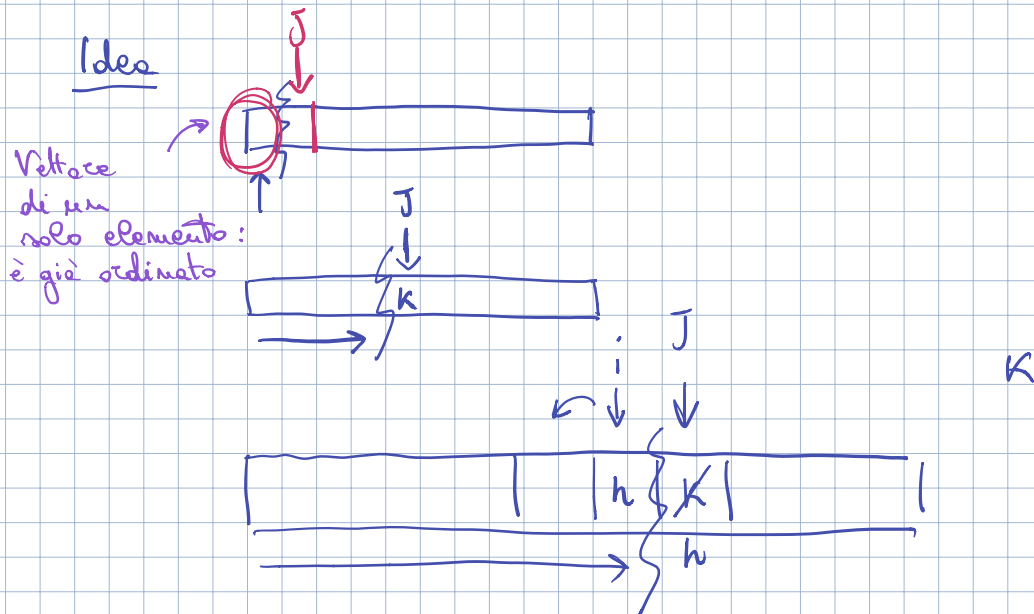
Idea

- Scendiamo il vettore, trovo il minimo e lo porto in prima posizione
- Itero sui restanti $n-1$ elementi

Esercizio per casa

Insertion Sort

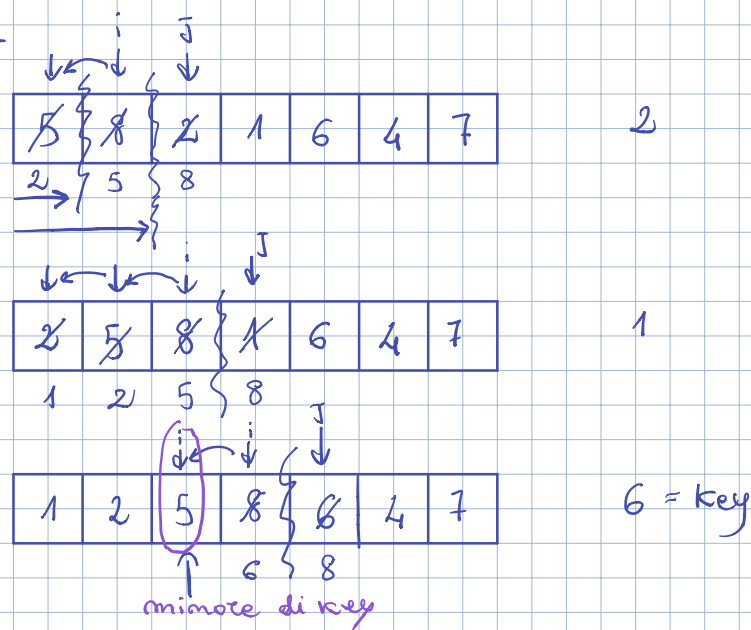
Idea



Se $A[i] \leq k$ ho finito \rightarrow incremento j

Se $A[i] > k$ sposto h a destra e i a sx

Esempio



```

InsertionSort (A) {
    for (j ← 2 to A.length) {
        i ← j - 1
        key ← A[j]
        while (i > 0 && A[i] > key) {
            A[i+1] ← A[i]
            i ← i - 1
        }
        A[i+1] ← key
    }
}

```