

Esempio di tema d'anno

Corso di Algoritmi e Linguaggi di Programmazione Python/C

Outline

- Parte1: Descrizione del problema
 - Il problema dell'Ordinamento
- Parte 2: Risoluzione del problema
 - Selection Sort – Descrizione dell'algoritmo
 - Diagramma di flusso ed analisi computazionale
- Parte 3: Conclusioni

Il problema dell'Ordinamento

- Un algoritmo di **ordinamento** (**sorting**) aiuta ad ordinare una lista
- Normalmente, si ragiona in termini numerici, dato che le liste di questo tipo sono le più semplici da ordinare
- È però possibile estendere il concetto a qualsiasi tipo di lista: sfruttando opportunamente diverse strutture dati, infatti, è possibile ricondursi al caso numerico, ed implementare un opportuno algoritmo di ordinamento

Selection Sort – Descrizione dell'algoritmo

- Il **selection sort** è un algoritmo **iterativo**
- La lista iniziale è suddivisa in due sottoliste
 - *Quella a sinistra rappresenta gli elementi già ordinati, quella a destra quelli non ancora ordinati*
- Ogni iterazione ha tre passi fondamentali
 - *Nel primo (che chiamiamo **setIndex**) è aggiornato l'indice dell'elemento attualmente analizzato*
 - *Nel secondo (che chiamiamo **findMin**) è individuato il valore minimo tra quelli non ordinati*
 - *Nel terzo (che chiamiamo **swap**) sostituiamo l'elemento attualmente analizzato con quello trovato nel primo step*

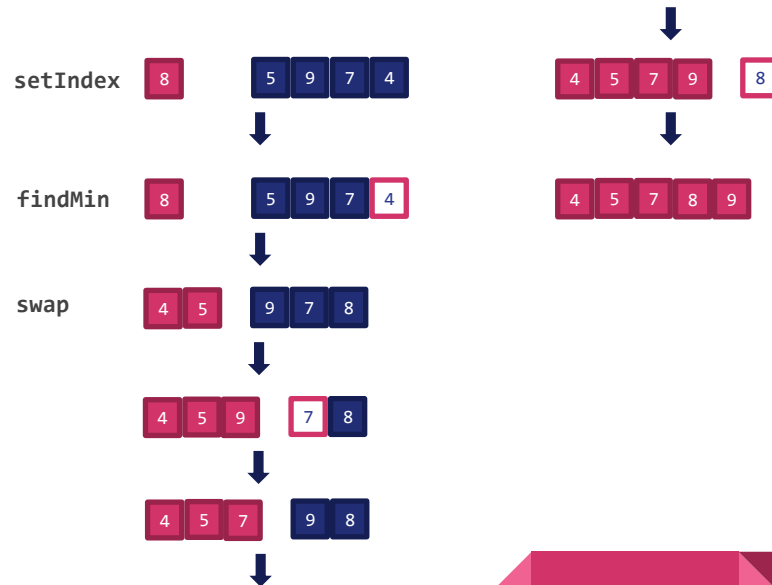
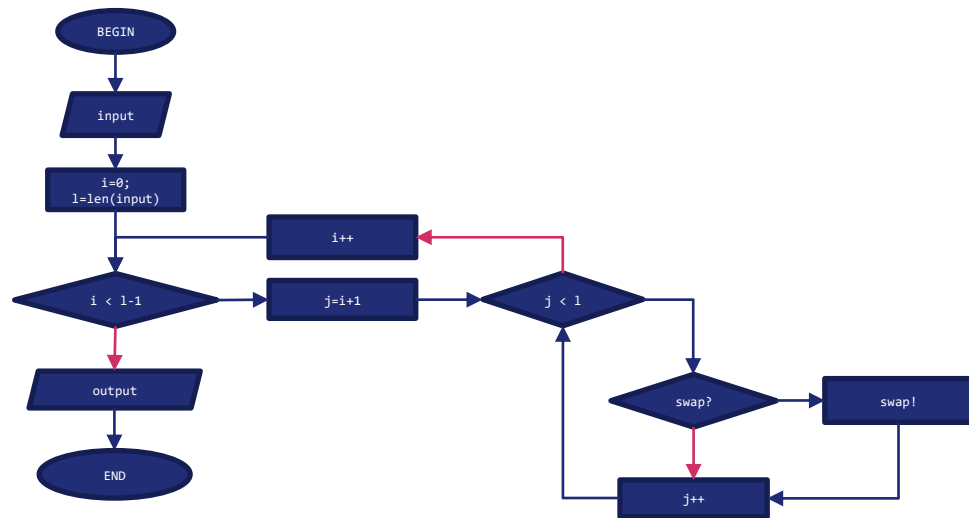


Diagramma di flusso ed analisi computazionale



- Il selection sort itera su tutti gli indici di un array
- Supponiamo un array di n elementi
 - Alla prima iterazione, avremo al più $(n - 1)$ confronti, ed uno swap
 - Alla seconda iterazione, avremo al più $(n - 2)$ confronti, ed uno swap, e così via
- Ciò significa che avremo bisogno al più di $n + (n - 1) + \dots + 1$ operazioni
- Questo valore equivale ad una serie aritmetica pari a

$$\frac{n^2}{2} + \frac{n}{2} \Rightarrow T \in O(n^2)$$

Conclusioni

- L'algoritmo di **selection sort** rappresenta il più semplice tra gli algoritmi di ordinamento dati.
- Dal punto di vista dell'implementazione è molto semplice, ma ha una complessità che cresce esponenzialmente con il numero di elementi all'interno del vettore da ordinare.
- Esistono metodi più avanzati, come merge sort, quick sort, etc.

Bonus: suggerimenti

- Cercate di suddividere adeguatamente l'effort tra i diversi membri del gruppo.
- Durante la presentazione, discuteremo prevalentemente della presentazione e del codice.
- Ricordatevi di consegnare il materiale da visionare in formato PDF.
- Ricordatevi di portare con voi una chiavetta contenente presentazione e codice.