

**Alcuni prompt per ChatGPT, utili per approfondimenti della lezione 4 di Programmazione I e Lab P I,  
(Lezioni P1-11-01, P1-11-03)**

**prof. Giunta**

Agisci come un tutor universitario di un corso di programmazione di primo livello. Descrivi l'algoritmo di ordinamento per inserimento (insertion sort). Scrivi una function C che implementa tale algoritmo.

Aiutami ad analizzare la complessità di tempo esatta di tale algoritmo con operazione dominante: confronto tra elementi dell'array. Poi descrivi tale complessità con la notazione asintotica, mostrandomi in dettaglio il ragionamento.

Aiutami ad analizzare la complessità di tempo esatta di tale algoritmo con operazione dominante: traslazioni a destra di un elemento dell'array. Poi descrivi tale complessità con la notazione asintotica, mostrandomi in dettaglio il ragionamento.

Descrivi l'algoritmo di ordinamento per selezione di massimo (selection sort). Scrivi una function C che implementa tale algoritmo. Utilizza una function `int max_ind(int a[], int n)` che restituisce l'indice dell'elemento massimo dell'array a di size n, e una function `void swap(int * x, int * y)` che scambia il valore delle variabili x e y.

Aiutami ad analizzare la complessità di tempo esatta di tale algoritmo con operazione dominante: confronto tra elementi dell'array. Poi descrivi tale complessità con la notazione asintotica, mostrandomi in dettaglio il ragionamento.

Descrivi l'algoritmo di ordinamento per selezione di minimo (selection sort). Scrivi una function C che implementa tale algoritmo. Utilizza una function `int min_ind(int a[], int n)` che restituisce l'indice dell'elemento minimo dell'array a di size n, e una function `void swap(int * x, int * y)` che scambia il valore delle variabili x e y.

Spiega in dettaglio questa istruzione `int minIndex = min_ind(a + i, n - i) + i;`

Discuti in modo comparativo la complessità di tempo degli algoritmi di insertion sort e di selection sort.

Approfondimento: Accenna a come l'algoritmo insertion sort potrebbe essere reso più efficiente introducendo l'idea del gap che porta all'algoritmo shellsort. Poi analizza le analogie tra l'algoritmo di ordinamento per inserimento e l'algoritmo shellsort

Approfondimento: Accenna a come l'algoritmo selection sort potrebbe essere reso più efficiente introducendo l'idea di memorizzare la parte non ordinata usando un heap (realizzato sempre nella porzione non ordinata dell'array) che porta all'algoritmo heapsort.