

**Alcuni prompt per ChatGPT, utili per approfondimenti della lezione 4 di Programmazione I e Lab P I,
(Lezioni P1-07-04, P1-07-06)**

prof. Giunta

Agisci come un tutor universitario di un corso di programmazione di primo livello. Scrivi una function C che determina il secondo massimo di un array 1D di float. Non usare l'allocazione dinamica. Analizza il numero di confronti dell'algoritmo.

Scrivi una void function C che determina contemporaneamente l'elemento massimo e l'elemento minimo di un array 1D. Analizza il numero di confronti dell'algoritmo.

Scrivi una void function C che determina contemporaneamente l'elemento massimo e l'elemento minimo di un array 1D. Analizza il numero di confronti dell'algoritmo.

Descrivi il problema della fusione di array ordinati (merging). Poi scrivi una function C che implementa l'algoritmo di fusione usando tre cicli while. Analizza il numero di confronti dell'algoritmo. Supponi che A=[33 35] e B=[2 4 6 8 22 23 28]. Quanti sono i confronti nel merge in questo caso?

Descrivi il problema di stabilire se due array 1D sono uguali, nel senso di avere uguali gli elementi di uguale posizione. Scrivi una logical function C per risolvere tale problema. Analizza il numero di confronti dell'algoritmo.

Descrivi il problema di stabilire se due array 1D sono uguali, nel senso di avere tutti uguali gli elementi indipendentemente dalla loro posizione (array equivalenti). Per esempio in questo senso [4 6 7] e [7 4 6] sono equivalenti. Scrivi una logical function C per risolvere tale problema. Analizza il numero di confronti dell'algoritmo. Usa una logical function appartiene che restituisce true se la chiave appartiene a un array e false altrimenti.

Nel caso in cui gli array in input non abbiano elementi duplicati e abbiano lo stesso size, come si potrebbe semplificare l'algoritmo precedente? Analizza anche in questo caso il numero di confronti.

Scrivi una function C che calcola la somma degli elementi che appartengono al triangolo superiore di una matrice. La diagonale principale appartiene al triangolo superiore. Considera prima il caso in cui la matrice è quadrata, poi il caso generale. Analizza il numero delle addizioni dell'algoritmo. Non usare +=

Riscrivi la function precedente usando la notazione a puntatore invece delle [[]].

Adesso considera il caso della somma degli elementi che appartengono al triangolo inferiore di una matrice. La diagonale principale appartiene al triangolo inferiore. Considera prima il caso in cui la matrice è quadrata, poi il caso generale. Non usare +=. Usa la notazione [[]]. Analizza il numero delle addizioni dell'algoritmo.

Riscrivi la function precedente usando la notazione a puntatore invece delle [[]].

Aiutami a risolvere il seguente esercizio: scrivere una function C che calcola il massimo elemento sul bordo di una matrice quadrata, cioè il massimo elemento dell'insieme costituito dalla prima riga, dalla prima colonna,

dall'ultima riga e dall'ultima colonna della matrice. Usa la notazione $[][]$. Analizza il numero di confronti dell'algoritmo.