

Laboratorio di programmazione strutturata (STM) - A.A. 2016-2017 -

Quarto appello - febbraio 2018

ESERCIZI

NON è consentito utilizzare libri o appunti.

Tempo a disposizione: 2 ore e mezza.

Esercizio 1 (18 punti)

Sia data una lista di interi.

1. Si definisca un tipo nodo per la lista con un campo a valori interi e un campo puntatore a nodo.
2. Si implementi una funzione in C `PorzioneLista`, che prende in input una lista A (data come un puntatore al suo primo nodo) di valori interi a_1, a_2, \dots, a_n e due interi positivi i e j . Questa deve restituire la sottolista di elementi consecutivi di A a partire dall' i -esimo fino al j -esimo escluso. Si adotta la convenzione che se i supera la lunghezza della lista oppure $j < i$ si restituisce la lista vuota mentre se $j > n$ viene restituita la sottolista da a_i fino all'ultimo elemento. Ad esempio se $A = [12, 14, 14, 15, 3, 3, 6, 8]$, `PorzioneLista(A, 2, 5)` deve restituire la nuova lista $[14, 14, 15]$ mentre `PorzioneLista(A, 1, 2)` deve restituire la lista $[12]$. Infine `PorzioneLista(A, 6, 10)` deve restituire $[3, 6, 8]$.

La funzione non dovrà utilizzare chiamate ad altre funzioni.

Si calcoli il costo della funzione definita, sia in termini di tempo che di memoria supplementare utilizzata rispetto alla lunghezza della lista in input.

Esercizio 2(12 punti)

1. Scrivere una funzione `TrovaMax` che prende come input un vettore V di dimensione n e una coppia di interi i ed f , $i \leq f \leq n$ e calcola l'indice del valore massimo degli elementi di V tra $V[i]$ e $V[f]$.
2. Scrivere una funzione `Ordina` per ordinare un vettore di interi che utilizza l'algoritmo selection-sort. La funzione `Ordina` dovrà richiamare la funzione `TrovaMax` del punto precedente.

Discutere la complessità di tempo di esecuzione delle due funzioni rispetto alla lunghezza del vettore.