Esame di programmazione del 28/8/2017 Turno 1

Data una lista concatenata L= 2->0->-1->3->0->1->2->10->2->-1, una sua sotto lista è una porzione contigua di L, come, per esempio: 3->0->1 (L= 2->0->-1->**3->0->1**->2->10->2->-1), 2->0->-1->3 (L= **2->0->-1->3**->0->1->2->10->2->-1) e 2->-1 (L= 2->0->-1->3->0->1->2->10->**2->-1**). Una sotto lista si dice crescente se ogni nodo ha campo info minore o uguale del successivo. Le 3 sotto liste di L menzionate prima non sono crescenti. La sotto lista -1->3 è crescente, anche se non è la più lunga sotto lista crescente di L che è 0->1->2->10. Per caratterizzare una tale sotto lista usiamo il tipo doppioN, vedi file dato, che ha 2 campi, *inizio* e *fine*, che puntano al primo e all'ultimo nodo della sotto lista ed un campo intero, *lung*, che indica la lunghezza della sotto lista stessa. Quindi per la sotto lista, 0->1->2->10, il valore doppioN che la rappresenta ha il campo *inizio* che punta al quinto nodo di L, il campo *fine* che punta all'ottavo nodo di L e *lung*=4.

Si chiede di scrivere una funzione iterativa ed una ricorsiva che data la lista L restituiscano il valore doppioN che rappresenta la sotto lista crescente più lunga in L. In caso di più sotto liste crescenti di uguale lunghezza e più lunghe di tutte le altre, si deve considerare quella che inizia prima in L.

Esercizio iterativo (6 punti): scrivere la funzione iterativa Fiter che soddisfa la seguente specifica:

PRE=(lista(L) corretta)

doppioN Fiter(nodo*L)

POST=(restituisce il valore doppioN che rappresenta la sotto lista crescente di lunghezza massima di L)

Esercizio ricorsivo (6 punti): si chiede di scrivere la funzione **ricorsiva** Frec che ha la stessa specifica di Fiter. La funzione Frec deve usare una funzione **ricorsiva** ausiliaria Aux che soddisfa la seguente specifica:

PRE=(lista(L) corretta e non vuota)

doppioN Aux(nodo*L)

POST=(restituisce il valore doppioN che rappresenta la sotto lista di L crescente di lunghezza massima che inizia col primo nodo di L)[1]

Seconda parte:

Data L e dato il valore doppioN prodotto da Fiter e da Frec, si chiede di scrivere Giter e Grec come segue:

Esercizio iterativo (5 punti): scrivere la funzione **iterativa** Giter deve rispettare la seguente specifica:

PRE=(lista(L) corretta, A di tipo doppioN individua una sotto lista di L, vL=L)

nodo* Giter(nodo*& L, doppioN A)

POST=(Giter restituisce col return la sotto lista di vL individuata da A e il valore di L diventa vL senza la sotto lista restituita col return)

Esercizio ricorsivo (5 punti): scrivere la funzione **ricorsiva** Grec che soddisfa le stesse specifiche di Giter.

Correttezza:

1) **(2 punti)** Scrivere l'invariante del ciclo principale di Fiter

2) **(2 punti)** Dare la dimostrazione induttiva di Aux

[1] Aux restituisce quindi il prefisso di L crescente di lunghezza massima. Visto che L è non vuota, esiste sempre un tale prefisso non vuoto.

3) **(2 punti)** Dare la dimostrazione induttiva di Frec.