

Esame di programmazione del 11/6/2018

L'esercizio è semplice: vogliamo ordinare una data lista concatenata e vogliamo farlo in modo ricorsivo e anche in modo iterativo. Vediamo un esempio.

Esempio. Se la lista è $L=4 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 0$, allora vogliamo riordinare i suoi nodi in modo che la lista diventi $0 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5$. Questo deve essere fatto usando i nodi della lista originale L. Nessun nuovo nodo va allocato oltre a quelli che formano L.

Esercizio ricorsivo. Si chiede di scrivere una funzione ricorsiva `ordric` che soddisfa la seguente specifica:

PRE=(lista(L) è ben formata, lista(ord) è ben formata, siano v_L e v_o i valori iniziali di lista(L) e lista(ord))

void `ordric`(nodo*&L, nodo*& ord)

POST=(L è vuota e lista(ord) è la lista ordinata composta dai nodi di v_L e di v_o)

Consiglio: conviene che `ordric` invochi una funzione ausiliaria che abbia lo scopo di inserire il prossimo nodo nella lista ordinata parziale ottenuta con i nodi di L considerati prima.

Esercizio iterativo. Si chiede di scrivere una funzione iterativa `orditer` che soddisfa la seguente specifica:

PRE=(lista(L) ben formata, v_L è il valore iniziale di lista(L))

nodo* `orditer`(nodo*L)

POST=(restituisce la lista composta dai nodi di v_L ordinati rispetto al campo `info`)

Consiglio: conviene che la funzione `orditer` usi una funzione (iterativa) ausiliaria che inserisca un nuovo nodo di L nella lista ordinata costruita con i nodi di L esaminati prima. E' anche utile usare una coda per gestire la lista ordinata man mano che essa è costruita. Per aiutarvi nel seguire questa idea, il file `dato` contiene la definizione di coda e le funzioni `push_beg` e `push_end` su una coda.

Correttezza :

- 1) Dare la prova induttiva della correttezza di `ordric`.
- 2) Specificare l'invariante del ciclo principale di `orditer`.