

Esame di Programmazione del 19/1/2017 V appello

L'esercizio usa le tre strutture dati, nodo, FIFOX e nodo_x che sono date nel programma reso disponibile nel moodle. In particolare, FIFOX serve a **gestire** una lista L di nodi di tipo nodo. In un valore FIFOX, i campi primo e ultimo puntano al primo e ultimo nodo della lista L, val_p e val_u sono i campi info del primo e dell'ultimo nodo di L ed n_ele è il numero di nodi di L. Il tipo nodo_x possiede un campo fix di tipo FIFOX* ed un campo next di tipo nodo_x*. Questo permette di costruire liste di nodi nodo_x in cui ogni nodo gestisce una lista di nodi di tipo nodo.

Il problema da affrontare è il seguente: dato un nodo_x N che gestisce una lista L di nodi di tipo nodo non vuota, e dato un intero $k \geq 1$, vogliamo dividere la lista L in tante sottoliste di k elementi (a parte l'ultima che potrebbe essere lunga meno di k). La prima di queste sottoliste verrà gestita dal nodo_x N (i cui campi andranno opportunamente modificati) mentre le altre sottoliste verranno gestite da nuovi nodi di tipo nodo_x. Tutti questi nodi nodo_x andranno concatenati in una lista ed N sarà sempre il primo nodo_x di questa lista.

Esempio. Consideriamo un nodo N di tipo nodo_x il cui campo fix punta al seguente FIFOX: primo e ultimo puntano al primo e ultimo nodo di questa lista (di tipo nodo), 3->2->4->1->0, poi val_p=3 e val_u=0 e n_ele=5. Quindi N gestisce la lista, 3->2->4->1->0.

Se $k=2$, vogliamo creare la lista di tipo nodo_x che inizia con N che gestisce la lista 3->2, poi N punta ad un nuovo nodo nodo_x che gestisce 4->1 e che punta ad un terzo nodo_x che gestisce la lista 0. Ogni nodo_x (anche N) deve avere tutti i campi definiti in modo coerente con la lista gestita dal nodo stesso.

Se $k=1$, allora verrà costruita una lista di 5 nodi nodo_x. Il primo dei 5 sarà N che gestirà la lista col solo nodo 3 e punterà ad un nuovo nodo_x che gestirà il solo nodo 2 così via, fino al quinto nodo_x che gestirà il solo nodo 0.

Se $k=5$ allora non succede nulla. Resta solo il nodo N senza cambiamenti. Lo stesso succede se $k>5$.

Programmazione (10+10): Si chiede di scrivere una funzione iterativa ed una funzione ricorsiva che eseguano l'operazione appena descritta. Queste 2 funzioni avranno prototipo e pre e postcondizione come segue.

PRE=(N gestisce una lista corretta di nodi di tipo nodo, chiamiamola L, N->next=0)

void partiziona_it(nodo_x* N, int k)

POST=(N è il nodo iniziale di una lista di tipo nodo_x tale che i nodi di questa lista gestiscano le sottoliste di L di lunghezza k (a parte l'ultima che può essere più corta) come descritto nell'esempio)

void partiziona_ric(nodo_x* N, int k) ha le stesse PRE e POST di partiziona_it.

Attenzione: Si consiglia di far sì che partiziona_it e partiziona_ric usino funzioni ausiliarie, ma esse devono essere, per partiziona_it solo iterative e per partiziona_ric solo ricorsive.

Correttezza (2+4+2):

- 1) Scrivere l'invariante del ciclo principale di partiziona_it.
- 2) Scrivere la prova induttiva di partiziona_ric.
- 3) Scrivere PRE e POST significative delle funzioni ausiliarie.