

Esame di programmazione del 18/6/2019 Parte Iterativa

Si tratta di scrivere una funzione iterativa preleva con il seguente prototipo:

```
coda preleva(nodo*&L, int*A, int nA)
```

che usa A come una sequenza di indici crescenti dei nodi della lista L e deve prelevare da L i nodi corrispondenti agli indici in A e restituirli in una coda, mentre i nodi rimasti devono venire restituiti nel parametro L (appunto passato per riferimento).

Esempio: sia $L = 2 \rightarrow 0 \rightarrow 3 \rightarrow -2 \rightarrow 10$ e $A = [1, 2, 4]$, la lista da prelevare è quella formata dai nodi di indice 1 (il secondo nodo), 2 (il terzo nodo) e 4 (il quinto nodo) e cioè $0 \rightarrow 3 \rightarrow 10$. Mentre la lista dei nodi che restano è $2 \rightarrow -2$. Se A contenesse anche altri valori, per esempio se fosse $A = [1, 2, 4, 6, 12]$, ovviamente i valori aggiunti non avrebbero effetto perché la lista L sarebbe finita dopo i primi 3 valori.

E' importante osservare che A contiene sempre elementi strettamente crescenti, visto che rappresentano indici di nodi successivi di L.

Vediamo un altro esempio. La lista L resta come prima, ma $A = [0, 1, 3]$, allora si devono prelevare i primi 2 nodi e poi il quarto, quindi $2 \rightarrow 0 \rightarrow -2$, mentre i nodi rimasti formano la lista $3 \rightarrow 10$.

Il caso appena presentato, cioè in cui si deve prelevare un prefisso di L (più, eventualmente, altri nodi), è quello più complicato. E' invece più semplice quando A non parte da 0, perché in questo caso in L resta un nodo che precede i nodi da prelevare.

Per questo motivo, in 3 test su 4 l'array A non inizia con 0. In questo modo un programma potrà passare 3 test anche senza trattare il caso che A parta con 0.

La PRE e POST di preleva sono le seguenti:

PRE= (Lista(L) è ben formata, A ha nA elementi strettamente crescenti, $vL = L$)

POST=(restituisce col return una coda che gestisce una lista con i nodi di vL che corrispondono agli indici in A, L è vL a cui sono tolti i nodi restituiti col return)

Correttezza : specificare l'invariante del ciclo che ritenete principale nella funzione preleva.

Attenzione: la struttura coda è definita nel programma dato che contiene anche la funzione push_end per inserire un nuovo nodo alla fine della lista gestita da una coda.