

Esercizio 1 del 21/5/2015 da consegnare per il 28/5

Si tratta di fare pattern matching di un array `int P[0..dimP-1]` sui campi info dei nodi di una lista concatenata `L`. Vediamo subito un esempio. Consideriamo solo match completi e contigui.

Esempio: sia `L=3->2->1->3->0->0->1` la lista, sia `dimP=3` e `P=[3,0,0]`, allora c'è un match di `P` nella lista `L` e inizia dal nodo di indice 3 della lista (gli indici dei nodi di una lista sono `0,1,2,...`). Si richiede in un caso di successo del match come questo, di restituire col `return` la parte della lista `L` in cui abbiamo trovato il match completo e contiguo, cioè la sottolista `3->0->0` di `L`, e attraverso il parametro `L`, passato per riferimento, si deve restituire quello che resta della lista `L` originale, cioè: `3->2->1->1`. La notazione che useremo per indicare queste 2 liste sarà: `match(L,P[3,0,0])` e `resto_match(L,P[3,0,0])`.

In caso `P` fosse `P=[3,0,1]`, non ci sarebbe match completo e contiguo in `L` e quindi col `return` si dovrebbe restituire `0` e col parametro passato per riferimento, andrebbe restituita la lista `L` originale. In questo caso `match(L,[3,0,1])=0` e `resto_match(L,[3,0,1])=L`. In caso ci fossero vari match di `P` in `L`, `match(L, P[0..dimP-1])` è quello che inizia prima in `L`.

Attenzione: nessun nodo diverso da quelli presenti in `L` va allocato. Nessun nodo va deallocato.

E' dato un main che compie le letture, fa le stampe e costruisce la lista `L` su cui tentare il match. Esso invoca la funzione ricorsiva `match` che deve rispettare il seguente prototipo ed essere corretta rispetto alle seguenti pre- e post-condizioni:

PRE=(`L` è una lista corretta, `P` ha `dimP` elementi, chiamiamo `vL=L`)

nodo* `match(nodo* &L, int*P, int dimP)`

POST=(la funzione restituisce col `return` `match(vL,P[0..dimP-1])` e `L=resto_mach(vL,P[0..dimP-1])`)

Consiglio: conviene introdurre anche almeno un'altra funzione ausiliaria che deve essere anch'essa ricorsiva.

Correttezza: scrivere pre- e post-condizione della funzione ausiliaria e poi dimostrare induttivamente la correttezza della funzione `match`.