Esercizio 2 del 25/5/2016

Si tratta di un pattern matching dove il pattern è un array int P[0..dimP-1] e il testo è una lista concatenata L(L1). Il matching che consideriamo deve essere completo, cioè l'intero P va trovato in L(L1), e deve essere anche contiguo. Quindi i diversi elementi di P devono venire trovati su nodi contigui di L(L1). Il main dato esegue tutte le letture necessarie per costruire L(L1) e per riempire P. Viene data anche la funzione match che ha il compito di scorrette L(L1) e per il nodo corrente invoca la funzione tenta che ha il compito di verificare se, a partire dal nodo corrente c'è un match di P completo e contiguo.

Vediamo alcuni **esempi:**

**(1)** supponiamo che L(L1)= 2->2-> 3-> 2-> 5-> 6-> 7-> 2-> 9-> 10-> 1 e P=[2,3,2], quindi dimP=3. C'è un match di P a partire dal secondo nodo di L(L1), quindi match dovrebbe restituire la lista corretta 2->3->2 col return mentre il parametro L1 passato per riferimento (e tale che L(L1)= 2->2-> 3-> 2-> 5-> 6-> 7-> 2-> 9-> 10-> 1) deve diventare: 2-> 5-> 6-> 7-> 2-> 9-> 10-> 1.

**(2)** Se manteniamo L(L1) come prima, cioè, 2->2-> 3-> 2-> 5-> 6-> 7-> 2-> 9-> 10-> 1 e P=[2,6], allora non c'è alcun match completo e contiguo e quindi match deve restituire 0 col return mentre L(L1) non cambia rispetto al valore iniziale.

La PRE e POST di match sono come segue:

PRE=(L(L1) è corretta, dimP>=0 e P ha dimP elem def., vL(L1)=L(L1))

POST=(se vL(L1)contiene un match di P, allora match restituisce col return una lista corretta che contiene i nodi del match di P che occorre più a sinistra, mentre L(L1) è la lista ottenuta da vL(L1) da cui sono tolti i nodi del match restituiti col return)&&(se vL(L1) non contiene match di P allora match restituisce 0 col return e L(L1)=vL(L1))

La funzione tenta invocata da match è da fare. Deve essere una funzione riorsiva e che soddisfa le seguenti PRE e POST:

PRE=(L(y) è lista corretta, dimP>=0, P[0..dimP-1] è def., vL(y)=L(y))

bool tenta(nodo*& y, int*P, int dimP, nodo*&m)

POST=(se i primi dimP nodi di vL(y) hanno campi info=P[0..dimP-1], allora tenta restituisce true, e L(y) è la lista corretta composta dai primi dimP nodi di vL(y) e m ha come valore la lista che resta da vL(y) una volta tolti i primi dimP nodi)&&(se i primi dimP nodi di vL(y) non esistono o non matchano P, allora tenta restituisce false e L(y)=vL(y)).

**Esempio 3:** In sostanza se invocassimo tenta con parametro attuale L1 ed il pattern dell'esempio (1) essa restituirebbe false (non c'è match che inizia dal primo nodo di L(y)) e la lista L(y) resterebbe inalterata. Se la invocassimo con L(y)= 2-> 3-> 2-> 5-> 6-> 7-> 2-> 9-> 10-> 1 e lo stesso P del caso (1), tenta restituirebbe true (c'è un match a partire dal primo nodo della lista L(y)), L(y) sarebbe 2->3->2 (corretta) e m=5-> 6-> 7-> 2-> 9-> 10-> 1.

**Attenzione**: tenta non cotruisce alcun nuovo nodo, né dealloca alcun nodo.

**Correttezza**: dimostrare la correttezza di tenta.