

Esercizio 2 del 7/6/2021

Come per l'esercizio 3 della settimana scorsa, si tratta di un esercizio di pattern matching in cui il pattern è, come di consueto, un array P di interi, ma il testo è un albero binario T. Si cerca il match di P sui cammini di T. In particolare si cerca un match contiguo e completo. Insomma se si trova il match di P[0] sul nodo r di T, allora si deve trovare il match di P[1] su un figlio di r così via per gli altri elementi di P e si deve riuscire a matchare tutto P lungo un solo cammino.

Esempio: sia $P=[12,4,3]$ e $T=10(12(4(_,3(_,_)),5(_,_)),7(_,6(_,0(_,_))))$. Si trova un match di P a partire dal figlio sinistro della radice, chiamiamolo fs. Da fs il match segue il cammino 0,1. Si noti che il match è completo (tutto P è matchato) e contiguo (il match parte da un nodo e continua su un suo figlio, poi un figlio di questo figlio).

Si chiede di scrivere una funzione PM1 che, nel caso il match sia trovato, costruisce una lista di nodi (di tipo nodoE) che puntano ai nodi di T in cui si è trovato il match. Il tipo nodoE (già visto la scorsa settimana) è come segue:

```
struct nodoE{ nodo* info; nodoE* next; nodoE(nodo* a=0, nodoE* b=0){ info=a; next=b; };
```

Per il match dell'esempio precedente, si deve costruire una lista con 3 nodi nodoE, dove il primo punta a fs, il secondo punta al figlio sinistro di fs e il terzo punta al figlio destro del figlio sinistro di fs. In generale si vuole produrre una lista di nodoE i cui nodi puntino ai nodi di T in cui si è trovato il match. Questa lista va costruita solo nel caso in cui si trovi un match completo e contiguo di P. Altrimenti la funzione deve restituire 0. Più PRECISAMENTE, chiede di scrivere una funzione ricorsiva PM1 che soddisfi alle seguenti specifiche:

PRE=(albero(T) ben formato, P contiene $\dim P \geq 0$ elementi)

nodoE* PM1(nodo*T, int*P, int dimP)

POST=(se in T esiste un cammino che contiene un match di P completo e contiguo, allora PM1 restituisce una lista di $\dim P$ nodi di tipo nodoE che puntano ai nodi del cammino più a sinistra su cui esiste un tale match di P, altrimenti PM1 restituisce 0).

Correttezza: dimostrare la correttezza della funzione PM1 rispetto alla PRE e POST date.

Attenzione: si consiglia di usare una funzione ausiliaria che si occupi di cercare un match di P che inizia in un certo nodo dell'albero. Questo è simile a quanto visto per il pattern match nelle liste concatenate. In quel caso, conviene avere una funzione ausiliaria che controlla se c'è un match di P che inizia in un dato nodo della lista.