

Esercizio 3 del 3/6/2021

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 chiede di attraversare T in ordine prefisso e questo significa fissare un ordine totale dei nodi di T e quindi P va cercato sui campi info di questa sequenza di nodi. Non si richiede che il match sia contiguo (cioè non è richiesto che P appaia in nodi contigui della sequenza che corrisponde al percorso prefisso) e neppure si richiede che il match sia completo (cioè, non è necessario che tutto P sia trovato). Basta matchare nei nodi di T (ordinati secondo il percorso infisso) il prefisso di P più lungo possibile.

Esempio: sia $P=[12,3,6,5]$ e $T=10(12(4(_,3(_,_)),5(_,_)),7(_,6(_,0(_,_))))$. Solo i primi 3 elementi di P si trovano in T: $P[0]=12$ si trova nel figlio sinistro (lo chiamiamo n_0) della radice di T, $P[1]=3$ si trova nel nodo a cui si arriva dalla radice seguendo il cammino $[0,0,1]$ (lo chiamiamo n_1). Infine $P[2]=6$ lo si trova nel nodo che corrisponde al cammino $[1,1,1]$ (lo chiamiamo n_2). Dopo n_2 , nell'ordine prefisso resta solo il nodo $[1,1,1,1]$ che non matcha $P[3]$. Quindi il match trovato non è completo. Inoltre non è contiguo perché l'ordine dei nodi di T secondo l'attraversamento prefisso sarebbe 10, 12, 4, 3, 5, 7, 6, 0 e i match sono quelli sottolineati che sono evidentemente non contigui (nell'ordinamento prefisso dei nodi). Si vuole costruire una lista costituita da nodi 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 n_0 , il secondo a n_1 e il terzo a n_2 . Insomma si vuole produrre una lista di nodoE i cui nodi puntino ai nodi di T in cui si è trovato un match e nell'ordine in cui questi nodi vengono trovati nell'attraversamento prefisso di T. Insomma prima n_0 , poi n_1 e infine n_2 .

Si chiede di scrivere una funzione ricorsiva PM che soddisfi alle seguenti specifiche:

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

nodoE* PM(nodo*T, int*&P, int &dimP)

POST=(se $n_0, n_1, \dots, n_{(k-1)}$, con $k \leq \dim P$, sono i nodi di T che matchano il massimo possibile di P percorrendo T in ordine prefisso, allora PM restituisce una lista di k nodoE che puntano a $n_0, \dots, n_{(k-1)}$).

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

Attenzione: osservare che P e dimP sono passate per riferimento.