

Esercizio 1 del 4/5/2016

E' lo stesso esercizio dell'esercizio 0 di questa settimana, ma anziché restituire il miglior match trovato, la funzione match deve restituire una lista concatenata i cui nodi contengono i valori M (dell'esercizio 0) che descrivono tutti i match non vuoti trovati a partire dalle diverse posizioni di T. Vediamo meglio con un esempio. I nodi della lista concatenata da costruire hanno il seguente tipo struttura:

```
struct nodoM {int lung, inizioT; nodoM* next; nodoM(int a=0, int b=-1, nodoM* c=0){lung=a; inizioT=b; next=c;}};
```

Esempio: sia $\dim P=3$, $P=[1,0,1]$, $\dim T=10$ e $T=[1,2,2,1,3,2,1,0,0,0]$. La lista concatenata da produrre è la seguente:

$[\text{lung}=1, \text{inizioT}=0] \rightarrow [\text{lung}=1, \text{inizioT}=3] \rightarrow [\text{lung}=2, \text{inizioT}=6]$ che indica che a partire dalla posizione 0 di T c'è un match del prefisso di P di lunghezza 1, poi, a partire dalle posizioni 1 e 2 di T non c'è match, a partire da 3 c'è un match di lunghezza 1, dalle posizioni 4 e 5 nessun match, dalla posizione 6 c'è un match di lunghezza 2 e a partire dalle posizioni 7, 8 e 9 nessun match.

Ovviamente se non c'è alcun match, la lista da produrre è vuota.

Cosa c'è da fare:

- a) viene dato un main che esegue l'i/o e invoca la funzione ricorsiva match del prossimo punto (b),
- b) va fatta una funzione ricorsiva match con il seguente prototipo e che deve essere corretta rispetto alle seguenti pre- e post-condizioni,

PRE= $(\dim P \geq 0, \dim T \geq 0, T[0..\dim T-1]$ è definita, $P[0..\dim P-1]$ è definita, $0 \leq iT \leq \dim T$)

$\text{nodoM}^* \text{ match}(\text{int}^* T, \text{int } \dim T, \text{int}^* P, \text{int } \dim P, \text{int } iT)$

POST=(restituisce la lista di tutti i match di prefissi di P in $T[iT..\dim T-1]$ che esistono, come descritto nell'esempio precedente)

La funzione match **deve** usare almeno un'altra funzione ricorsiva.

Correttezza: Dare una dimostrazione induttiva della correttezza di match.