Esercizio 0 del 4/5/2016

Si tratta di un esercizio ricorsivo in cui si vuole fare pattern matching di un array int P[0..dimP-1] in un array int T[0..dimT-1]. A differenza di altri esercizi già visti di pattern matching, ora consideriamo anche il caso di match di prefissi di P. Quindi ci interessa il match dell'intero P[0..dimP-1], ma anche di P[0], di P[0..1], di P[0..2] eccetera e vogliamo trovare il match di massima lunghezza.

Esempio: se dimP=5 e P=[1,0,1,2,1], allora oltre al match di tutto P, dobbiamo considerare anche quello di [1,0,1,2] (i primi 4 elementi di P), e quello di [1,0,1] (i primi 3 elementi) e di [1,0] (i primi 2 elementi di P) e di [1] (il primo elemento di P). Insomma dobbiamo considerare tutti i prefissi di P. Un prefisso di P ha una lunghezza. Per esempio [1,0,1] ha lunghezza=3, mentre [1,0] ha lunghezza=2.

Per un certo prefisso di P, per esempio [1,0,1] un suo match in T è una porzione T[i,i+1,i+2] di T che sia identica a [1,0,1]. Un tale match ha inizio in T nella posizione i. Quindi useremo la seguente struttura (con costruttore) per rappresentare un match di un prefisso di P in T:

struct M {int lung, inizioT; M(int a=0, int b=-1){lung=a; inizioT=b;}};

Nell'esempio precedente di un match di [1,0,1] a partire da T[i], esso sarà rappresentato dalla struttura M con lung=3 e inizioT=i.

Si tratta di scrivere una funzione ricorsiva che determini il match di un prefisso di P in T di lunghezza massima e che produca il valore di tipo M che rappresenta questo match. In caso ci siano match diversi con la stessa lunghezza, si chiede quello con inizioT massimo.

Esempio: sia dimP=5, P=[1,0,1,2,1], dim=10 e T=[1,1,0,1,3,2,1,0,0,0]. Il match di lunghezza massima è quello del prefisso [1,0,1] di P che è rappresentato dal valore M= [lung=3, inizioT=1]. Questo valore M deve essere scritto su cout dal programma richiesto. Se T=[1,1,0,1,3,2,1,0,1,0], allora [1,0,1] avrebbe 2 match caratterizzati da [lung=3, inizioT=1] e da [lung=3, inizioT=6] ed è il secondo che andrebbe stampato perché ha inizioT maggiore a parità di lung . Esiste anche il caso in cui per nessun prefisso (non vuoto) di P esista un match su T (succede se P[0] non compare in T). In questo caso il programma deve stampare [lung=0, inizioT=-1].

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=(dimP>=0, dimT>=0, T[0..dimT-1] è definita, P[0..dimP-1] è definita, 0<=iT<=dimT)

M match(int*T, int dimT, int*P, int dimP, int iT)

POST=(restituisce un valore M che rappresenta il match di lunghezza massima in T[iT..dimT-1]di un prefisso di P e, a parità di lunghezza massima va restituito quello con inizioT massimo, qualora nessun prefisso abbia match, la funzione deve restituire il valore M [lung=0,inizioT=-1])

c) il main stampa il valore M campo per campo. Provate a ridefinire l'operatore << per il tipo M in modo da poter sostituire la stampa del main con un semplice << ottenendo lo stesso risultato.

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

Correttezza: è richiesta la pre- e post-condizione delle funzioni ausiliarie. La correttezza di match va dimostrata induttivamente.

Consiglio: non preoccupatevi, almeno in prima istanza, se la vostra soluzione sembra piuttosto inefficiente. Cercate prima di tutto la semplicità.