

Corso di programmazione 2009-2010

Quarta esercitazione per casa: assegnata il 2 febbraio 2010

Consegna: il 9 febbraio entro le 8 di mattina, con il comando: **consegna settimana4**

Introduzione: in un problema di pattern matching abbiamo un testo T , per esempio $\text{char } T[1000]$, ed un pattern P (per esempio, $\text{char } P[\text{dim_P}]$) ed esaminiamo T da sinistra a destra cercando una porzione di dim_P elementi contigui di T che siano identici agli elementi di P . Più precisamente, cerchiamo un k tale che $T[k..k+\text{dim_P}-1] == P[0..\text{dim_P}-1]$. Quando questo si verifica diremo di avere trovato un **match** di P in T e diremo che questo match **comincia nella posizione k di T** . Due match di P in T , che iniziano nelle posizioni k_1 e k_2 di T , sono **sovrapposti**, se gli intervalli $[k_1, k_1+\text{dim_P}-1]$ e $[k_2, k_2+\text{dim_P}-1]$ hanno qualche indice in comune e altrimenti sono detti **non sovrapposti**.

Dei 2 esercizi che seguono, solo l'(1) è richiesto, mentre il (2) è opzionale (per chi ci ha preso gusto). Per l'esercizio (1) si richiede di specificare la pre- e la postcondizione di ciascuna funzione e di associare un invariante a ciascun ciclo.

(1) Si chiede di scrivere una funzione $\text{char } (*)[10][15] \text{ F}(\text{char } (*T)[10][15], \text{int } \text{limite1}, \text{char } P[], \text{int } \text{dim_P})$ che soddisfi i seguenti punti:

(i) limite1 è il limite della prima dimensione di T , cioè T è un array $\text{char } T[\text{limite1}][10][15]$ ed è completamente definito;

(ii) Per ogni strato $T[i]$ e per ogni sua riga $T[i][j]$, F deve calcolare il numero N_j di match di P (anche sovrapposti) sulla riga. Poi deve calcolare la somma $N_0+N_1+\dots+N_9=M_i$, M_i è il **valore** dello strato $T[i]$. E' importante osservare che un match su una riga deve essere completamente contenuto sulla riga. Non può continuare alla riga successiva;

(iii) F deve restituire il puntatore allo strato di T che ha valore massimo (in caso ci siano più strati a valore massimo, basta restituire il puntatore ad uno qualsiasi);

(2) Si tratta di scrivere una funzione $\text{char } (*)[10][15] \text{ G}(\text{char } (*T)[10][15], \text{int } \text{limite1}, \text{char } P[], \text{int } \text{dim_P})$ per i cui parametri vale quanto detto in (1) e che soddisfi i seguenti punti:

(i) Per ogni strato $T[i]$, G deve calcolare il numero di match di P **non sovrapposti** che esistono su $T[i]$. Si osservi che, a differenza dell'esercizio (1), qui un match può iniziare su una riga $T[i][j]$ e continuare sulla riga successiva $T[i][j+1]$, a condizione che $j+1 < 10$, cioè che la riga $j+1$ sia ancora una riga dello strato $T[i]$; il numero totale di questi match è il **valore** dello strato $T[i]$;

(ii) G deve restituire il puntatore allo strato di valore massimo (uno qualsiasi se ce n'è più di uno)

NOTA: è consentito definire anche altre funzioni oltre F e G .