

Esercizio 5 del 4/11/2014

L'esercizio riguarda il pattern matching, come per l'esercizio 4 e adotteremo la stessa terminologia dell'esercizio 4: chiameremo T l'array di interi che costituisce il testo su cui cercare il match, T ha $\dim T$ elementi, mentre P è il pattern ed ha $\dim P$ elementi. La lettura da cin di $\dim T$, $\dim P$, e degli elementi di T e P è come per l'esercizio 4.

Nell'esercizio 4 il pattern P veniva cercato in posizioni contigue di T, cioè un match è un sotto-array $T[i..i+\dim P-1]$ di lunghezza $\dim P$ di T che è esattamente identico a $P[0..\dim P-1]$. Nel presente esercizio il match non è necessariamente contiguo in T, quindi un match è costituito da $\dim P$ indici $0 \leq i_0 < i_1 < \dots < i_{(\dim P-1)} < \dim T$, tale che, per ogni j in $[0..\dim P-1]$, $T[i_j] = P[j]$, e inoltre, se $0 \leq j < \dim P-1$, allora $i_{(j+1)} \leq (i_j) + 2$, cioè l'indice $i_{(j+1)}$ (che è l'indice del match che viene subito dopo i_j) è o $(i_j) + 1$ oppure è $(i_j) + 2$.

Esempio: se i_j fosse 4, allora $i_{(j+1)}$ può essere solo 5 o 6. Nel caso in cui $i_{(j+1)}$ fosse 6, diremo che c'è un mismatch.

E' facile capire che, se gli indici $i_0, \dots, i_{(\dim P-1)}$ individuano un match, allora il valore $MIS = (i_{(\dim P-1)} - i_0 + 1) - \dim P$ è il numero dei mismatch del match. Ovviamente $MIS \leq \dim P - 1$. Nel seguito per indicare un match con eventuali mismatch (e al più un mismatch tra i match di elementi successivi di P), useremo il termine "match con 1 mismatch" (M1MIS).

Il programma richiesto, dopo le letture di $\dim T$, $\dim P$, T e P, dovrà cercare l'M1MIS di P in T che ha minimo MIS e dovrà restituire l'indice in cui questo M1MIS ha inizio (cioè i_0) e il suo valore MIS. In caso ci fossero più M1MIS che hanno valore di MIS minore o uguale a tutti gli altri M1MIS, si deve considerare il M1MIS con i_0 minimo.

Esempio: sia $\dim T = 12$, $\dim P = 3$ $T = [0, 2, 3, 1, 0, 2, 2, 1, 2, 1, 2, 2]$ e $P = [1, 2, 2]$. Ci sono diversi M1MIS. Il primo da sinistra inizia in posizione 3 di T, in 4 c'è un mismatch e poi la coppia di 2 nelle posizioni 5 e 6. Quindi questo M1MIS ha $MIS = 1$. Il secondo M1MIS inizia in posizione 7, continua in 8, in 9 ha un mismatch e di conclude in 10. Di nuovo questo M1MIS ha $MIS = 1$. Tra questo e il precedente, dovremmo restituire il precedente visto che inizia in una posizione più piccola di T (inizia in 3 anziché in 7). Ma c'è un terzo M1MIS che inizia in posizione 9 e continua in 10 e 11. Quindi il suo MIS è 0 ed è quindi il migliore M1MIS. Quindi il programma richiesto per questo esempio dovrebbe stampare su cout 9 e 0.

PRE=(cin contiene $\dim T$ e $\dim P$ ($0 < \dim T \leq 100$ e $0 < \dim P \leq 20$), seguiti da $\dim T + \dim P$ valori interi qualsiasi)

POST=(se c'è almeno un M1MIS di P in T, il programma stampa l'inizio del M1MIS a MIS minimo (e quello che inizia prima se ce ne fossero diversi minimi) e il valore del suo MIS) && (se non esiste M1MIS di P in T allora il programma scrive su cout "nessun M1MIS")

Correttezza: si chiede di scrivere un invariante per ciascun ciclo del programma. Si chiede anche di dimostrare la correttezza del ciclo più interno, se c'è annidamento di cicli e di un ciclo qualsiasi se non c'è annidamento, con la prova in 3 parti vista in classe.