

Esercizio di Programmazione dell'8 settembre 2014

Il problema da considerare è lo stesso di quello dell'esame del 18/7/2014. Viene letta da "input" una sequenza di interi come la seguente: 2 3 -1 0 2 3 -1 2 3 -1 0 2 -1 -1 2 3 0 -1 -2

si tratta di una sequenza di sotto-sequenze, ognuna terminante con la sentinella -1, mentre la fine della sequenza è segnalata dalla sentinella -2. La lunghezza di ciascuna sotto-sequenza non tiene conto del -1 finale. Quindi la sotto-sequenza vuota è costituita solo da -1 ed ha lunghezza 0. E' possibile che "input" contenga anche solo -2 che rappresenta la sequenza vuota. Il main (dato) legge la sequenza di interi contenuta in "input" nell'array C fino al -2 finale, compreso.

Si vuole costruire un array X di coppie, dove ogni coppia corrisponde ad una delle sotto-sequenze contenute in C nel modo seguente: la prima componente della coppia è l'indice di C in cui inizia la sotto-sequenza corrispondente, mentre la seconda componente è la lunghezza della sotto-sequenza stessa (senza contare il -1).

Esempio 1. L'array X che corrisponde alla sequenza data prima è: 0 2 3 3 7 2 10 2 13 0 14 3
quindi per esempio la prima coppia mostra che la prima sotto-sequenza ha lunghezza 2 (ovviamente essendo la prima, parte dall'indice 0 di C), mentre la coppia (10,2), che è la quarta coppia, indica che la quarta sotto-sequenza di C inizia in posizione 10 ed ha lunghezza 2. La coppia (13,0) mostra che in C c'è una sotto-sequenza vuota che inizia nella posizione 13 di C.

Parte ricorsiva:

Si chiede di scrivere una funzione ricorsiva IC che calcoli X (nel compito del 18/7 si chiedeva invece di farlo con una funzione iterativa). Si chiede anche che l'array X sia allocato dinamicamente e IC sia in grado di allungarlo di 20 posizioni quando il numero di interi da inserire in X eccede la dimensione corrente di X. Il main (dato) crea X di 20 posizioni (10 coppie), mentre l'allungamento (di 20 posizioni alla volta) deve essere gestito all'interno della funzione ricorsiva IC. IC ha il seguente prototipo e deve soddisfare la seguente coppia di pre- e post-condizioni:

PRE_IC=(C è definito e a partire dall'indice nC contiene m sotto-sequenze di interi, con $m \geq 0$, ognuna terminante con -1, alla fine delle quali c'è la sentinella -2, lim è un multiplo di 10, X ha $\text{lim} * 2$ elementi con $0 \leq nX \leq \text{lim}$, sia $nX = v_nX$, $v_lim = \text{lim}$)

void IC(int *C, int nC, int*& X, int& nX, int & lim)

POST_IC=(X[v_nX*2..nX*2-1] contiene m coppie di valori che corrispondono (come spiegato nell'Esempio 1) alle m sotto-sequenze contenute in C dalla posizione nC fino al -2 finale, $0 \leq nX \leq \text{lim}$, $\text{lim} \geq v_lim$ è un multiplo di 10 e X ha $\text{lim} * 2$ elementi)

Parte iterativa:

Si chiede di scrivere una funzione iterativa che calcoli l'array Z che ha tanti elementi quante sono le sotto-sequenze in C (e le coppie di interi in X). Per la sotto-sequenza S_i di indice i, $Z[i]$ è il numero sotto-sequenze che S_i contiene (senza contare S_i stessa). La definizione di inclusione è la seguente:

Definizione di inclusione: Una sotto-sequenza S contiene tutti i suoi prefissi, anche la sotto-sequenza vuota.

Esempio 2. Se $S = [3 \ 2 \ 4 \ -1]$, S contiene [3], [3 2], [3 2 4] e anche [-1] (che, essendo vuota, è contenuta in ogni sotto-sequenza).

La funzione iterativa richiesta, deve seguire il seguente prototipo e pre- e post-condizioni:

PRE_M=(C contiene $nX \geq 0$ sotto-sequenze e X contiene nX coppie che descrivono le sotto-sequenze nel modo richiesto nella parte ricorsiva, Z ha nX elementi tutti a 0)

M(int* C, int *X, int nX, int*Z)

POST_M=(per j in $[0.. nX-1]$, Z[j] è il numero delle sotto-sequenze in C che la sotto-sequenza di indice j contiene (senza contare se stessa))

Correttezza:

- a) dimostrare con una prova induttiva che IC è corretta rispetto a PRE_IC e POST_IC date.
- b) specificare invarianti per i cicli usati in M. Mostrare che si tratta effettivamente di invarianti.

Avvertenze:

- a) è possibile introdurre funzioni ausiliarie in più rispetto alle funzioni IC ed M, richieste espressamente. Quelle della parte ricorsiva devono essere ricorsive (con l'eccezione di operazioni semplici che non richiedano né iterazione né ricorsione) e quelle della parte iterativa devono essere iterative.
- b) nella correzione della parte iterativa si premieranno le soluzioni che evitino di confrontare tra loro più volte le stesse coppie di sotto-sequenze. Per una qualsiasi coppia di sotto-sequenze un solo confronto (ben fatto) dovrebbe essere sufficiente per arrivare a calcolare l'array Z giusto.

Per le integrazioni:

Scrivere anche una versione iterativa della funzione IC. Il prototipo può essere diverso, ma la funzione deve calcolare X, nX e lim come richiesto. Non fare la parte di correttezza.