Esercizio 2 del 17/2 consegna il 24/2

Compitino del 13/2/2014 Turno 2

Viene dato un main che apre i file "input" e "output", dichiara 2 array T di 200 interi e P di 20 e poi legge i seguenti valori da "input":

1) n e poi n valori che vengono inseriti nelle prime n posizioni di T, 0<n<=200;
2) legge lim1, lim2 e lim3, tutti maggiori di 0, **si deve assumere che lim1*lim2*lim3<=200**;
3) dimP, 0<dimP<=20, e poi legge dimP valori nelle prime dimP posizioni di P;
4) per finire legge n_m, maggiore di 0.

Poi il main invoca la funzione:

void trova_strato(int *T, int n, int lim1, int lim2, int lim3, int* P, int dimP, in n_m, int& indice_strato)

La funzione trova_strato è da fare in modo che soddisfi le seguenti pre- e post-condizioni:

PRE=(T ha i primi n elementi definiti, n<=200, P ha i primi dimP elementi definiti, dimP<=20, n_m>0)

POST=(se vediamo T come un array a 3 dimensioni X[lim1][lim2][lim3], allora indice_strato ha per R-valore l'indice minimo di uno strato di X che contiene una riga che contiene esattamente n_m match di P[0..dimP-1], **NON sovrapposti tra loro**; se nessuno strato soddisfa questa condizione allora indice_strato=-1)

La nozione di match sovrapposti tra loro è illustrata nell'esempio seguente. Si osservi che dalla condizione (2) sull'input segue che **X non avrà necessariamente tutti gli elementi definiti**. E' importante capire che l'array X è solo una maniera di "vedere" l'array ad una dimensione T. **Non sono accettate soluzioni in cui l'array X è effettivamente dichiarato e T è ricopiato in X.**

Esempio: Supponiamo che n=32, lim1=5, lim2=4 e lim3=6 e che i seguenti siano i 32 valori letti in T:

1 0 1 0 1 1 1 0 1 1 2 0
1 0 2 1 1 1 1 0
1 1 1 0 1 1
1 0 1 1 0 1

mostriamo i 32 valori come apparirebbero in X. Supponiamo ora che dimP=3 e P=[1,0,1] e n_m=2. La riga 0 dello strato 0 ha 2 match di P, ma questi sono sovrapposti tra loro (il terzo numero del primo match è anche il primo del secondo match). Quindi la riga 0 non soddisfa la condizione della POST. Lo stesso è vero per la riga di indice 2 che contiene un solo match, mentre n_m=2. La riga di indice 3 dello strato 0 invece ha 2 match non sovrapposti e quindi soddisfa la POST. Visto che la POST chiede l'indice minimo di uno strato con una riga con n_m match non sovrapposti, a questo punto, la funzione trova_strato dovrebbe terminare assegnando 0 alla variabile indice_strato e senza neppure guardare lo strato 1.

Consigli: conviene introdurre delle funzioni ausiliarie oltre a trova_strato. Cercate sempre di evitare operazioni inutili.

Correttezza: Associare ad ogni ciclo un invariante ed una post-condizione. Dimostrare che la vostra funzione trova_strato è corretta rispetto alla PRE e POST date prima. Se usate funzioni ausiliarie, associate loro pre- e post-condizioni che descrivano cosa fanno.