

## Esercizio 1 del 17/2. Consegna il 24/2

Compitino del 13/2/2014 Turno 1

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,  $0 < n \leq 200$ ;
- 2) legge lim1, lim2 e lim3, tutti maggiori di 0, **si deve assumere che  $n \leq \text{lim2} * \text{lim3}$** ;
- 3) dimP,  $0 < \text{dimP} \leq 20$ , 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_colonna(int *T, int n, int lim1, int lim2, int lim3, int* P, int dimP, int n_match, int& indice_c)
```

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

PRE=(T ha i primi n elementi definiti,  $n < \text{lim2} * \text{lim3}$ , P ha i primi dimP elementi definiti,  $n\_m > 0$ )

POST=(se vediamo T come un array a 3 dimensioni  $X[\text{lim1}][\text{lim2}][\text{lim3}]$ , allora indice\_c ha per R-valore l'indice minimo di una colonna di X che contiene **esattamente** n\_m match di  $P[0..\text{dimP}-1]$ , anche sovrapposti tra loro; se nessuna colonna soddisfa questa condizione allora indice\_c=-1)

Si osservi che dalla condizione (2) sull'input segue che **X avrà solamente il primo strato con elementi definiti e che questo strato non sarà necessariamente completamente definito**. 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  $n=23$ ,  $\text{lim1}=5$ ,  $\text{lim2}=5$  e  $\text{lim3}=6$  e che i seguenti sono i 23 valori letti in T:

```
0 0 1 0 1 1
1 2 2 1 1 1
1 1 2 1 1 1
0 1 1 2 1
```

mostriamo i 23 valori come apparirebbero nel primo strato di X (l'unico strato con elementi definiti).

Supponiamo ora che  $\text{dimP}=2$  e  $P=[1,1]$  e  $n\_m=2$ . C'è una sola colonna che presenta **esattamente** 2 match di  $[1,1]$  ed è la colonna di indice 5 che è composta da 3 uni. Si noti che i 2 match sono sovrapposti: il secondo match condivide l'1 nella riga 1 con il primo match. La colonna 0 ha 1 solo match, mentre la colonna di indice 4 ne ha 3, quindi non soddisfano la POST che ne richiede esattamente  $n\_m=2$  (in questo esempio). Quindi indice\_c deve ricevere il valore 5. Si osservi che P in generale sarà costituito da valori qualsiasi e differenti tra loro.

**Consigli:** è utile introdurre qualche funzione ausiliaria oltre a trova\_colonna. Cercate sempre di evitare operazioni inutili.

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