

Correzione compitino del 17/11/2014

Il parte

annunci:

- video

- regole degli esami

- uso dei file

- esercizio 1, LB1, LB2 e tra poco LB3

- ore d'ufficio lunedì 14:30 – 16 (TA 4° piano, stanza 404)

L'esercizio richiede di considerare le ripetizioni all'interno delle righe di T (considerando solo gli elementi definiti). Si vuole calcolare l'indice m della riga che contiene il massimo numero di ripetizioni di uno stesso valore n . La tripla (m,n,k) rappresenta il fatto che nella riga m il valore n ripete k volte.

La tripla (m,n,k) è massima tra quelle esistenti per T se per ogni altra tripla (j,q,p) , vale che: o $p < k$ oppure, se $p = k$, allora $q < n$ oppure, se $p = k$ e $q = n$, allora $j > m$.

Il programma deve calcolare la tripla massima tra quelle esistenti per T .

NOTARE: nella slide precedente la frase:

La tripla (m,n,k) è massima tra quelle esistenti per T se per ogni altra tripla (j,q,p) , vale che: o $p < k$ oppure, se $p = k$, allora $q < n$ oppure, se $p = k$ e $q = n$, **allora $j > m$.**

è uguale a quella del turno 1 del compitino del 17/11/2014. Purtroppo, durante la lezione di oggi, nonostante abbia inizialmente dato le spiegazioni giuste sulla condizione $j > m$, durante lo sviluppo del programma ho erroneamente considerato che la condizione fosse $j < m$. La confusione è dovuta al fatto che questa è la condizione adottata nell'esercizio del turno 2.

Pertanto, la soluzione presentata nelle slide successive è corretta per $j > m$ ed è quindi **diversa** da quella presentata a lezione e che potete rivedere nel video. **Le differenze sono contrassegnate da commenti in verde e si trovano nelle slide 8 e 9.** Quindi avete le soluzioni sia per la versione con $j > m$ (nel seguito) che con $j < m$ (nella lezione registrata).

Sono spiacente della confusione.

per ogni riga vogliamo calcolare qual'è l'elemento che ripete più volte

quindi è un problema di massimo su ciascuna riga e poi tra righe diverse

se c'è un elemento che ripete (su una riga) più di tutti gli altri, è facile

in caso di parità si confrontano i valori ripetuti e in caso di ulteriore parità si preferisce la riga minore.

dietro a tutto questo il solito problema della eventuale riga incompleta

la scansione di tutte le righe (anche l'eventuale riga più corta)

```
int briga,bval, brip=0, i=0, rt=nrp;  
if(rem>0) rt++;  
while(i<rt) //R=(0<=i<=rt)&&(briga, bval, brip sono la massima tripla per  
le righe 0..i-1)  
{int lung=10;  
if(i==nrp) lung=rem; // i=nrp=> riga i è la riga più corta di lung=rem
```

CORPO= confronta le ripetizioni degli elementi della riga....
}

- a).....calcoliamo tripla migliore per la riga e poi la confrontiamo con la migliore sulle righe precedenti ?
- b)oppure confrontiamo ogni ripetizione con la tripla migliore?

CORPO caso (a)

```
int j=0, cval, crip=0;
while(j<lung) //R=(0<=j<=lung)&&(cval e crip meglio in
T[i][0..j-1])
{
    int v=T[i][j], k=0, rip=0;
    while (k<lung) //R=(0<=k<=lung)&&(rip = N. occ. di v in
T[i][0..k-1])
    {if (T[i][k]==v) rip++;
        k++;
    }
    if(rip>crip || (rip==crip && v>cval))
    {crip=rip; cval=v;}
    j++;
} //POST =(cval e crip = meglio della riga i)
```

CICLO sulle RIGHE

```
while(i<rt) //R=(0<=i<=rt)&&(briga, bval, brip sono la
massima tripla per le righe 0..i-1)
{int lung=10;
if(i==nrp) lung=rem;
//
CORPO (slide precedente)
//POST =(cval e crip = meglio della riga i)
if(crip>brip || (crip==brip && cval>bval)
{brip=crip; bval=cval; briga=i;} //in caso crip==brip &&
cval==bval , briga<i e quindi briga non cambia, vedi slide 4
i=i+1;
} //POST=(brig, bval e brip massima tripla nelle righe 0..rt
(ciascuna col giusto numero di elementi))
```


CORPO caso (b)

```
int j=0; //non serve crip e cval
while(j<lung) //R=(0<=j<=lung)&&(bval e brip e brig
meglio nelle righe 0..i-1 e in T[i][0..j-1])
{
    int v=T[i][j], k=0, rip=0;
    while (k<lung) //R=(0<=k<=lung)&&(rip = N. occ. di v in
T[i][0..k-1])
    {if (T[i][k]==v) rip++;
        k++;
    }
    if(rip>brip || (rip==brip && v>bval) ) //diverso da lezione
    {brip=rip; bval=v; brig=i;}
    j++;
} //POST =(bval brip e brig = meglio delle righe 0...i)
```

CICLO sulle RIGHE caso (b)

while(i<rt) //R=(0<=i<=rt)&&(briga, bval, brip sono la
massima tripla per le righe 0..i-1)

```
{int lung=10;
```

```
if(i==nrp) lung=rem;
```

CORPO (slide precedente)

//POST =(bval brip e brig = meglio delle righe 0...i) che
mostra che il successivo test non serve più

```
if(crip>brip || (crip==brip && cval>bval)
```

```
{brip=crip; bval=cval; briga=i;}
```

```
i=i+1;
```

```
} //POST=(brig, bval e brip massima tripla nelle righe 0..rt  
(ciascuna col giusto numero di elementi))
```