Elementi comuni

Si progetti un algoritmo che determina se c'è almeno un elemento in comune tra due array ordinati di interi. L'algoritmo deve dare in output la coppia di indici di un elemento comune, la coppia (-1,-1) altrimenti. Se n è il numero degli elementi del primo array e m quello del secondo, l'algoritmo dovrebbe terminare in O(n+m).

Elementi comuni - Soluzione

Siano A,B i due array ordinati in input.

Si modifica la funzione fondi per fare in modo che dia in output la coppia di indici di un elemento comune.

Nella funzione fondi gli elementi nei due array sono confrontati a due a due e l'elemento più piccolo viene selezionato per essere copiato nel nuovo array, poi si confronta l'elemento successivo al selezionato con quello dell'altro array.

In questa modifica invece se due elementi sono uguali si esce dal ciclo dei confronti dando in output la coppia di indici trovata.

Si continua la ricerca scartando il più piccolo tra i due elementi confrontati, infatti questo non può essere presente nell'altro visto che tutti gli elementi successivi a quello del confronto sono più grandi di lui.

Il tempo complessivo di esecuzione è in O(n+m)

Esempio

L[0] > R[0] quindi R[0] è minore di tutti gli elementi sucessividi L incrementiamo j
L[0] > R[1] quindi R[1] è minore di tutti gli elementi successivi di L incrementiamo j
L[0] < R[2] quindi L[0] è minore di tutti gli elementi successivi di R incrementiamo i
L[1] < R[2] quindi L[1] è minore di tutti gli elementi successivi di R incrementiamo i
L[2] = R[2] uscita, output (2,2)

Elementi comuni: pseudocodice

```
ElComune(A,B)
input: due array
prec: gli elementi di A e di B sono ordinati in ordine crescente
output: dà in output  la coppia di indici che individuano il primo elemento in comune, (-1,-1) altrimenti.
/è una versione modificata della funzione fondi, utilizzata per fondere due array ordinati in un unico array ordinato.
i=0, j=0
n= A.length
m= B.length
while (i < n and j < m) do
        if (A[i] == B[j]) return (i,j)
        if  (A[i] < B[j]) then i++  (in tal caso A[i]<B[j] ≤ B[j+1]≤ …≤B[m-1]  quindi A[i] non è presente in B)
        else j++    (in tal caso B[j]<A[i] ≤ A[i+1]≤ …≤A[n-1] quindi B[j] non è presente in A)
if (i == n or  j == m) then return (-1,-1)
( se i == n  e j ≤ m-1 allora A[n-1] < B[j] ≤ B[j+1]≤ …≤B[m-1]
        quindi nemmeno A[n-1] è presente in B,
e se j == m  e i ≤ n-1 allora B[m-1] < A[i] ≤ A[i+1]≤ …≤A[n-1]
        quindi nemmeno B[m-1] è presente in A)
```