

**CORSO DI
ALGORITMI E STRUTTURE DATI**
Prof. ROBERTO PIETRANTUONO

Indicazioni

Si consegna un file in **formato editabile (.txt, .docx, .rtf, etc.)** nominandolo “*CognomeNome*”, in cui è riportata l’implementazione (nel linguaggio scelto) seguita da una indicazione della complessità temporale dell’algoritmo implementato (complessità nel caso peggiore, è sufficiente il limite superiore $O(f(n))$). Se si utilizzano librerie di cui non si conosce la complessità, lo si indichi nella spiegazione (ad esempio, “la complessità è $O(n \log n)$ al netto della complessità dell’algoritmo x , che è non nota”). Se si utilizza la randomizzazione, si indichi anche il tempo di esecuzione atteso.

PROBLEMA

Sia data una sequenza di eventi numerati da 1 a n , in ordine arbitrario, che definiamo sequenza di riferimento. Siano date delle sequenze di input di pari dimensioni, n . Si implementi un algoritmo che, per ogni sequenza di input, determini la sotto-sequenza di elementi (non necessariamente contigui) più lunga i cui elementi rispettano lo stesso ordine relativo della sequenza di riferimento. Ad esempio, se la sequenza di riferimento è 4 2 3 1, allora per la sequenza di input 2 3 4 1 la sotto-sequenza più lunga che rispetta l’ordine relativo degli elementi della sequenza di riferimento è 2, 3, 1 (gli elementi infatti appaiono nella sequenza di input nello stesso ordine relativo in cui appaiono nella sequenza di riferimento).

INPUT

La prima riga dell’input contiene il numero di casi di test.

Ciascun caso di test inizia con una riga che riporta il numero n di elementi della sequenza con $2 \leq n \leq 10$. Seguiranno n righe, ciascuna con n numeri interi. La prima di tali righe rappresenta la sequenza di riferimento. Le restanti $n-1$ righe rappresentano le sequenze di input. Il caso di test termina con una riga con il solo numero 0.

OUTPUT

Per ogni caso di test, si stampi dapprima la stringa “TEST OUTPUT”, seguita da $n-1$ righe ciascuna riportante il risultato (ossia la sotto-sequenza più lunga) per ogni sequenza di input. Si noti che per una sequenza di input possono esistere più sotto-sequenze corrette a lunghezza massima; in tal caso, se ne stampi una qualsiasi (ad esempio per la sequenza di riferimento 1 2 3 4, la sequenza di input 1 3 2 4 contiene due sotto-sequenze a lunghezza 3, che sono 1 2 4 ed 1 3 4).

Sample Input

```
2
4
4 3 2 1
1 2 3 4
3 1 2 4
4 3 2 1
0
6
1 2 3 4 5 6
```

1 2 3 4 6 5

6 3 5 1 2 4

5 4 6 2 1 3

2 4 1 3 5 6

1 2 6 5 4 3

0

Sample Output

TEST OUTPUT

1

2

4

TEST OUTPUT

5

3

2

4

3