

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

**Indicazioni**

Si consegna un file in **formato .txt** nominandolo *CognomeNome.txt*, 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 1**

Data una matrice di  $N$  righe e  $M$  colonne, si scriva un algoritmo per trovare la minima differenza assoluta tra uno qualsiasi dei due elementi adiacenti di un array di dimensione  $N$  creato selezionando un elemento da ogni riga della matrice (l'elemento selezionato dalla riga 1 sarà  $arr[0]$ , l'elemento selezionato dalla riga 2 sarà  $arr[1]$  e così via). Una soluzione a forza bruta prevede la costruzione di tutti i possibili array, che avrebbe complessità  $O(M^N)$ . Si implementi una soluzione a complessità non esponenziale (si suggerisce di utilizzare dapprima un algoritmo di ordinamento sulle righe).

**INPUT**

L'input è costituito da una prima riga che indica il numero di casi di test  $T$ . Le righe successive sono i casi di test. Ogni caso di test inizia con una riga che indica la dimensione di riga e colonna, e segue poi una matrice che rappresenta il labirinto  $L$ .

**OUTPUT**

Si riporti per ogni caso di test il valore della differenza minima.

**Sample Input**

```
1
2 2
8 5
6 10
```

**Sample Output**

```
1
```

**PROBLEMA 2**

Sia data in input una stringa di caratteri. Si implementi un algoritmo di **backtracking** per stampare tutte le possibili stringhe che possono essere create posizionando uno spazio oppure nessuno spazio tra i caratteri. Ad esempio con "XYZ" posso ottenere: XYZ, XY Z, X YZ, X Y Z.

**INPUT**

L'input è costituito da un numero intero  $T$  che indica il numero di casi di test, seguito da  $T$  linee contenenti ciascuna un caso di test (ossia una stringa di input).

## OUTPUT

Per ogni input, l'algoritmo stampi l'elenco di tutte le possibili stringhe che si possono ottenere. Al termine dell'output per ogni caso di test si stampi **"\*END\*"**.

### Sample Input

2  
XYZ  
ABCDE

### Sample Output

XYZ  
XY Z  
X YZ  
X Y Z  
\*END\*  
ABCDE  
ABCD E  
ABC DE  
ABC D E  
AB CDE  
AB CD E  
AB C DE  
AB C D E  
A BCDE  
A BCD E  
A BC DE  
A BC D E  
A B CDE  
A B CD E  
A B C DE  
A B C D E  
\*END\*