

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

Si consideri un labirinto L dato come matrice binaria $N \times N$ di blocchi in cui il blocco sorgente è il blocco più in alto a sinistra, cioè $L[0][0]$ e il blocco di destinazione è il blocco più in basso a destra, cioè $L[N-1][N-1]$. Un topo parte da $L[0][0]$ e deve raggiungere la destinazione. Il topo può muoversi solo in due direzioni: in avanti e in basso. Nella matrice, 0 significa che il blocco è un vicolo cieco e 1 significa che il blocco può essere utilizzato nel percorso. Si scriva un algoritmo di Backtracking per determinare un percorso dalla sorgente alla destinazione. Il percorso è scritto in una nuova matrice S delle stesse dimensioni di L , in cui la cella è '1' se fa parte del percorso, 0 altrimenti.

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 il numero del caso di test e, a seguire, la matrice risultato S per quel caso di test.

Sample Input

```
1
4 4
1 0 0 0
1 1 0 1
0 1 0 0
1 1 1 1
```

Sample Output

```
1
1 0 0 0
1 1 0 0
0 1 0 0
0 1 1 1
```