

Progettazione di Algoritmi - appello giugno1 2020

Esercizio 1. [Conta] Il display di un telefonino si presenta come di seguito indicato:

1	2	3
4	5	6
7	8	9
*	0	#

Cerchiamo un particolare numero telefonico e sappiamo che:

- il numero è composto da n cifre.
- non contiene cifre uguali adiacenti
- nel comporre il numero sul tastierino basta spostarsi solo tra tasti adiacenti in orizzontale o verticale

Ad esempio, per $n = 7$, la combinazione 12108586996 non è di certo il numero telefonico che cerchiamo a causa della presenza delle seguenti tre coppie di cifre adiacenti 10 e 86 e 99.

Progettare un algoritmo che, dato n , restituisce il numero di combinazioni possibili per il numero telefonico da ricercare.

Ad esempio:

- per $n = 1$ la risposta dell'algoritmo deve essere 10
- per $n = 2$ la risposta dell'algoritmo deve essere 26.
(i numeri possibili sono infatti: 08, 12, 14, 21, 23, 25, 32, 36, 41, 45, 47, 52, 54, 56, 58, 63, 65, 67, 74, 78, 80, 85, 87, 89, 96, 98).

L'algoritmo deve avere complessità $O(n)$. Motivare bene la correttezza e la complessità dell'algoritmo proposto.

Esercizio 2. [Enumera] Abbiamo una matrice M di interi di dimensione $n \times n$ con $n > 1$. Una *discesa* su questa matrice è una sequenza di n celle della matrice con i seguenti vincoli

- le celle appartengono a righe diverse della matrice
- la prima cella appartiene alla prima riga della matrice
- ogni altra cella è adiacente (in verticale o in diagonale) alla cella che la precede.

Ad esempio, per $M = \begin{pmatrix} 12 & 10 & 3 & 14 & 9 \\ 0 & 1 & 13 & 15 & 13 \\ 8 & 10 & 1 & 2 & 7 \\ 7 & 11 & 10 & 5 & 7 \\ 18 & 4 & 6 & 10 & 0 \end{pmatrix}$

sono evidenziate due possibili discese (12, 1, 1, 11, 4 e 3, 15, 2, 5, 6).

Progettare un algoritmo che, data la matrice M , stampa tutte le possibili discese di M .

Ad esempio per $M = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ l'algoritmo stamperà le 4 discese: 1, 3 1, 4 2, 3 2, 4.

L'algoritmo deve avere complessità $O(n^2 S(n))$ dove $S(n)$ è il numero di discese da stampare.

Motivare bene la correttezza e la complessità dell'algoritmo proposto.