

## ESERCIZI PD

1) Supponete di avere una scacchiera con  $n \times n$  caselle e una pedina. Quando posizionata sulla generica casella  $(i,j)$  per la pedina sono possibili al più due mosse:

- spostarsi verso il basso nella casella  $(i+1, j)$ , se  $i < n-1$ ;
- spostarsi verso destra nella casella  $(i, j+1)$ , se  $j < n-1$ .

Progettare un algoritmo di programmazione dinamica che calcola il numero di percorsi possibili per spostare la pedina dalla casella  $(0, 0)$  in alto a sinistra alla casella  $(n-1, n-1)$  in basso a destra (ad esempio, in una scacchiera  $3 \times 3$  i percorsi possibili sono 6). Valutare la complessità dell'algoritmo proposto.

2) Una pedina è posizionata sulla casella  $(0, 0)$  in alto a sinistra di una scacchiera  $n \times n$  e deve raggiungere la casella  $(n-1, n-1)$  in basso a destra. Quando posizionata sulla generica casella  $(i, j)$  per la pedina sono possibili al più due mosse:

- spostarsi verso il basso nella casella  $(i+1, j)$ , posto che  $i < n-1$
- spostarsi verso destra nella casella  $(i, j+1)$ , posto che  $j < n-1$ .

La pedina raggiungerà dunque la casella  $(n-1, n-1)$  con un cammino che la porterà a toccare  $(2n-2)$  caselle. Ad ogni casella della scacchiera è inoltre associato un valore  $c_{ij}$ , ed il valore del cammino è dato dalla somma dei valori delle caselle toccate dalla pedina.

Progettare un algoritmo che trova il cammino di valore massimo in  $O(n^2)$ .

3) Supponete di avere una scacchiera con  $n \times n$  caselle e una pedina che dovete muovere dall'estremità inferiore a quella superiore. Una pedina si può muovere una cella in alto, oppure una cella in diagonale destra, oppure una cella in diagonale sinistra. Le celle sono denotate da una coppia di coordinate  $(x,y)$ . Quando una cella  $(x,y)$  viene visitata, guadagnate  $p(x,y) \geq 0$ .

Calcolare un percorso da una qualunque casella dell'estremità inferiore ad una qualunque casella dell'estremità superiore, massimizzando il profitto. Definire anche una procedura che stampi il percorso calcolato.

4) Si consideri il problema di calcolare l'*Edit Distance* tra le due stringhe  $P = \text{CACCIUCCO}$  e  $T = \text{COCCIUTO}$ , considerando il peso di un errore di *mismatch* = 2, e quello di una inserzione o cancellazione = 1. Simulare l'algoritmo di programmazione dinamica per il calcolo dell'edit distance mostrando il contenuto della tabella che l'algoritmo riempie dinamicamente.

5) Si consideri il problema dell'allineamento globale (*Edit Distance*) tra la sequenza  $P = \text{SARTA}$  e la sequenza  $T = \text{LASTRA}$ , considerando il peso di un errore di *mismatch* 1, quello di una inserzione o cancellazione 1 e quello di un'*inversione di caratteri* 0.5. Per esempio facendo corrispondere la coppia di caratteri SA della prima stringa con AS si incorre in un errore di tipo inversione.

Si progetti un algoritmo di programmazione dinamica per trovare il punteggio del miglior allineamento globale e si simuli sulle stringhe date.