

008AA – ALGORITMICA E LABORATORIO
Secondo Compitino, 1 Giugno 2017

Cognome Nome:

N. Matricola:

Corso: A B

Esercizio 1. [7+2 punti]

Supponete di avere una scacchiera con $n \times m$ caselle e una pedina. Quando posizionata sulla generica casella (i, j) , la pedina può muoversi in tre soli modi:

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

Progettare un algoritmo, basato sulla programmazione dinamica, che calcola il numero di percorsi possibili per spostare la pedina dalla casella $(1, 1)$ in alto a sinistra alla casella (n, m) in basso a destra (ad esempio, in una scacchiera 3×3 i percorsi possibili sono 13). Valutare la complessità dell'algoritmo proposto.

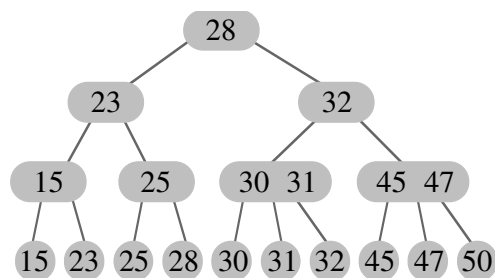
Esercizio 2. [6+4 punti]

Dato un grafo **orientato** $G = (V, E)$ con n vertici e m archi, un vertice *bersaglio* $t \in V$ e un sottoinsieme $V' \subset V$ contenente k vertici, progettare un algoritmo per individuare il vertice di V' con la minima distanza dal bersaglio t . L'algoritmo deve restituire uno qualunque dei vertici a distanza minima, oppure NIL se il bersaglio non è raggiungibile da nessun vertice di V' . Si supponga di avere in input, oltre al grafo G rappresentato con liste di adiacenza, il bersaglio t e un array binario S tale che $S[i] = 1$ se il vertice i appartiene a V' , 0 altrimenti.

1. Progettare un algoritmo di costo in tempo $T(n, m, k) = O(k(n + m))$.
2. Progettare un algoritmo di costo in tempo $T(n, m, k) = O(n + m)$ che risolve il problema con una sola visita. [**Suggerimento:** è ammessa la modifica del grafo G in input all'algoritmo]

Esercizio 3. [3+3 punti]

Si consideri l'albero 2-3 T mostrato in figura.



1. Disegnare l'albero 2-3 T_1 ottenuto da T rimuovendo il nodo contenente la chiave 25.
2. Disegnare l'albero 2-3 T_2 ottenuto da T_1 inserendo un nodo contenente la chiave 12.

Esercizio 4. [2+3 punti]

Definire il *problema dell'arresto* e dimostrarne l'indecidibilità.