1. Consider an $m \times m$ chessboard with an integer written on each cell of the board. A pawn starts out in the first column and first row and advances one row at a time to the last row. To advance, the pawn may move forward one row along a column or a diagonal. Thus a pawn in row $i$, column $j$, can move to columns $j - 1$, $j$, or $j + 1$ in row $i + 1$. The weight of a move is given by the integer on the cell to which the pawn moves. The weight of a sequence of moves is the sum of the weights of each move (including the weight of cell $(1,1)$).

Describe an algorithm which computes the minimum weight sequence of moves by which a pawn can advance from the first to the last row. (It will simplify your solution if you introduce two additional columns, column 0 and column $n + 1$, with an infinite weight on each cell.)

2. Let $A = a_1a_2\ldots a_m$ and $B = b_1b_2\ldots b_n$ be two strings of characters. We want to transform $A$ into $B$ using following operations:

- delete a character
- add a character
- change a character

Write a dynamic programming algorithm that finds the minimum number of operations needed to transform $A$ into $B$.

3. $N$ jobs are to be scheduled for processing on one machine. Job $i$, $1 \leq i \leq N$, needs $t_i$ units of processing time. If job $i$ is finished by time $T$, where $T$ is a given deadline, then a profit $p_i$ is earned; otherwise, a penalty $q_i$ is imposed. (Both $p_i$ and $q_i$ are positive integers.) We want to select a subset $S$ of jobs such that

(i) $\sum_{i \in S} t_i \leq T$, and
(ii) $f(S) = \sum_{i \in S} p_i - \sum_{i \notin S} q_i$ is maximum.

Show how to find such a set of jobs using dynamic programming.