CMPT 307 - Data Structures and Algorithms:
Problem Set 3

For each of the following problems, give a dynamic-programming solution, and give
the worst-case time analysis of your algorithm (using the asymptotic notation). Your solution
should follow the 4-step format: (i) describe an array and say how to obtain an optimal
value from your array; (ii) give a recurrence for your array, and prove its correctness; (iii)
give a pseudo-code algorithm for filling in the array; and (iv) give an algorithm for finding
an optimal solution to the original problem, using the values in the array (if such an optimal
solution exists).

1. **Making Change** You’re given an unlimited supply of coins of denominations $d_1, d_2, \ldots, d_k$
   (where each $d_i$ is a positive integer), and a positive integer number $n$ (all in binary).
   Your goal is to give the amount $n$ of money, using in total as few coins as possible.

   More formally, you need to design an algorithm for the following problem.
   Given: positive integers $d_1, d_2, \ldots, d_k$ and $n$ (in binary).
   Find: a sequence of nonnegative integers $v_1, v_2, \ldots, v_k$ (where each $v_i$ is the number of
   coins of denomination $d_i$) so that $\sum_{i=1}^{k} v_i d_i = n$, and $\sum_{i=1}^{k} v_i$ is minimized.

   (For example, for denominations 100, 25, 10, 5, 1, the value 57 is best given using coins
   $25 + 25 + 5 + 1 + 1$, for the total of 5 coins.)

2. **String Matching** Given $n, m \in \mathbb{N}$ and binary strings $s, t_1, \ldots, t_m$, where $n$ is the
   length of $s$, decide whether $s$ can be expressed as a concatenation of some of the
   strings $t_j$ (allowing repetitions), and if so, output such a representation of $s$. (For
   example, if $n = 8$, $m = 3$, $s = 01110101$, $t_1 = 000$, $t_2 = 01$, $t_3 = 11$, then the answer is
   YES since $s = t_2 t_3 t_2 t_2$, and $\langle 2; 3; 2; 2 \rangle$ is the representation.)

3. **Maximum Weight Independent Set of a Path** A path on $n$ nodes is a graph
   where all edges are of the form $(i, i + 1)$, for $i = 1, \ldots, (n - 1)$. With each node $i,
   1 \leq i \leq n$, we associate a positive integer weight $w_i$.

   The problem Max Weight Independent Set of a Path is: Given a path graph on $n$
   nodes, with node weights $w_1, \ldots, w_n$, find an independent set $S \subseteq \{1, \ldots, n\}$ of the
graph (where no two nodes in $S$ are connected by an edge) such that the weight $\sum_{i \in S} w_i$ of $S$ is maximized.

(For example, given a path on 5 nodes 1, 2, 3, 4, 5, with the corresponding weights 1, 8, 6, 3, 6, the max weight independent set is $S = \{2, 5\}$ of the total weight $8 + 6 = 14$. So your algorithm must return $S = \{2, 5\}$.)

4. Also, the following three problems from the textbook [KT], Chapter 6: #3, #8, #11.