COMP 3711 – Spring 2019
Tutorial 6b

1. Give an $O(n^2)$ time dynamic programming algorithm to find the longest monotonically increasing subsequence of a sequence of n numbers, i.e, each successive number in the subsequence is greater than or equal to its predecessor.

   For example, if the input sequence is $\langle 5, 24, 8, 17, 12, 45\rangle$, the output should be either $\langle 5, 8, 12, 45\rangle$ or $\langle 5, 8, 17, 45\rangle$.

2. The subset sum problem is: Given a set of n positive integers, $S = \{x_1, x_2, \ldots, x_n\}$ and an integer W determine whether there is a subset $S' \subseteq S$, such that the sum of the elements in S' is equal to W.

   For example, if $S = \{4, 2, 8, 9\}$ and $W = 11$, then the answer is "yes" because there is a subset $S' = \{2, 9\}$ whose elements sum to 11. If $W = 7$. the answer is "no".

   Give a dynamic programming solution to the subset sum problem that runs in $O(nW)$ time. Justify the correctness and running time of your algorithm.

3. The (Restricted) Max-Sum Problem.

   Let A be a sequence of n numbers $a_1, a_2, \ldots, a_n$.

   Find a subset S of A that has the maximum sum, provided that, if $a_i \in S$, then $a_{i-1} \notin S$ and $a_{i+1} \notin S$.

   Note that, unlike in the previous question, A is a sequence in which order matters (and not an unordered set).

   As an example, if $A = 1, 8, 6, 3, 7$, the max possible sum is $S = \{8, 7\}$.