CMSC451 Homework 5
Spring 2017 due April 10, 2017 at 3pm EDT

For all algorithms, provide time complexity analysis as well as a formal proof of correctness. Homework solutions should be clearly written and electronically submitted via ELMS (`http://elms.umd.edu`) by the due date listed above.

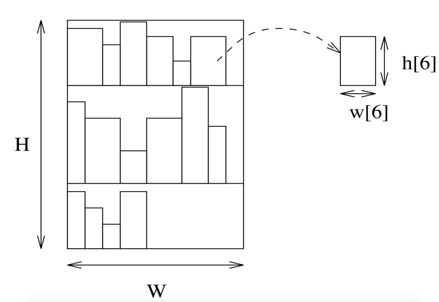
1. Suppose you are given a collection of n books that must be placed on a single library bookcase. Let $h[1..n]$ and $w[1..n]$ be arrays, where $h[i]$ denotes the height of the i-th book and $w[i]$ denote its width. Assume that these arrays are given by the books' catalog numbers and that the books must be shelved in this order, e.g. there are no books shelved between books i and $i+1$. Also assume that books are stacked in the usual upright manner; you cannot lay a book on its side. This bookcase has fixed width W, but you may make it any height H you like, you may use as many shelves as you would like, and you may put as many books on each shelf as you like (up to the width of the bookcase). Assume for simplicity that shelves take up no vertical height.

   The *book shelver's problem* is, given $h[1..n], w[1..n]$ and W, to find the minimum height bookcase layout that can shelve all the books. Below is an example of a layout. The height and width of the 6th book are shown on the right. Notice that there is some wasted height on the third shelf.

   (a) Write down a recurrence for the book shelver's problem.

   (b) Use your recurrence to develop an efficient dynamic programming algorithm to solve this problem.

   (c) As always, give proof of correctness and runtime analysis for your algorithm.



2. You are planning a long road trip from DC to Los Angeles and wish to find the shortest route to LA, but are not willing to spend more than C dollars in tolls.

   We can model this problem as follows: given is a directed graph $G = (V, E)$, where each edge e has a *length* $\ell_e > 0$ and a *cost* $c_e > 0$. Also, two vertices $s \in V$ and $t \in V$ are specified, as well as a budget of C dollars. We wish to find the shortest length path from vertex s to vertex t, subject to the total (cumulative) *cost* of the path being within C dollars. Assume that the cost of each edge is an integer.

Design an algorithm to find this path with running time $O(f(n, m)\ C)$ where $f(n, m)$ is a polynomial function in n, the number of vertices, and m, the number of edges in the network. Assume that the directed graph G is represented as an adjacency list.

3. Recall that in the Floyd-Warshall algorithm, matrix D_i contains the shortest-path costs using intermediate nodes of label at most i. Prove that the Floyd-Warshall algorithm still works, even if D_i is stored in the same matrix that stored D_{i-1} (i.e., the entries of D_i clobber their respective entries in D_{i-1}).

4. KT Chapter 6, Exercise 1 (page 312).