Reminder that project checkpoint 1 is also due next week, on Friday 10/5 at 12pm. Thus this set has only one full design question.

1. 4.9

2. Full write-up for (c). You are working in the warehouse of an online store and are given the task to decide how to best stack a set of \( n \) boxes. Each of the \( b_i \) (1 \( \leq \) \( i \) \( \leq \) \( n \)) box has an associated weight \( w_i \). Customers may purchase merchandises contained in any of the boxes at any time, which will require you to retrieve the corresponding box. You are told that the probability of sale for each box \( b_i \) is \( p_i \), where \( 0 \leq p_i \leq 1 \), and \( \sum_{i=1}^{n} p_i = 1 \). In addition, since the boxes are stacked on top of each other, all boxes on top of the desired one must be lifted off before it can be accessed.

A stacking of boxes can be specified by a permutation \( S = \langle s_1, \ldots, s_n \rangle \) of the numbers \( \{1, \ldots, n\} \). Given a stacking \( S \), the individual cost of getting to the \( i \)th box is the product of its sales probability and the sum of weight of all boxes above it and itself, that is, \( C_i(S) = p_{s_i} \cdot \left( \sum_{j=1}^{s_i} w_j \right) \). The total cost of a stacking \( S \) is the sum of all individual costs, \( T(S) = \sum_{i=1}^{n} C_i(S) \).

For example, given the following 4 boxes:

\[
\begin{align*}
    b_1 : & \quad w_1 = 300 \quad p_1 = 0.4 \\
    b_2 : & \quad w_2 = 200 \quad p_2 = 0.35 \\
    b_3 : & \quad w_3 = 500 \quad p_3 = 0.1 \\
    b_4 : & \quad w_4 = 100 \quad p_4 = 0.15
\end{align*}
\]

The following stacking of \( S = \langle s_1 = 4, s_2 = 2, s_3 = 1, s_4 = 3 \rangle \) results in a total cost of

\[
T(S) = 0.15 \cdot 100 + 0.35 \cdot (100 + 200) + 0.4 \cdot (100 + 200 + 300) + 0.1 \cdot (100 + 200 + 300 + 500) = 470
\]

You may disregard any concerns of ceiling height, as well as dimensions and mechanical strengths of the boxes. In other words, assume that all permutations give stacking orders that are physically reasonable.

(a) Present a (short) counterexample to show that stacking the boxes in increasing order (from top to bottom) of weight \( w_i \) is not optimal

(b) Present a (short) counterexample to show that stacking the boxes in decreasing order of access probability \( p_i \) is not optimal

(c) Present an algorithm, which given a list of \( w_i \) and \( p_i \), determines a stacking \( S \) of minimum total cost. As always, justify your algorithm’s correctness and derive its running time.
3. Consider placing \( n \) books on shelves in a library. The order of the books is fixed by the catalog numbers and cannot be changed. Each book \( b_i \) has a thickness \( t_i \). The length of each bookshelf is \( L \). The heights of the books can be ignored since any book fits on any shelf. A greedy algorithm places books in order \( \{b_1, \ldots, b_n\} \) starting at shelf 1. If the \( i \)th book fits on the current shelf, it is placed there, if not, it is placed on the next shelf, and the process continues like this until all books are placed. Let \( k_j \) denote the total thickness of the books placed on shelf \( j \), and assume a total of \( m \) shelves were used.

1. Would the described greedy algorithm minimize \( \max_{1 \leq j \leq m} (L - k_j) \) (the maximum unused length over \( m \) shelves)? If it does, give a proof. If it does not, give a counter example.

2. Would the described greedy algorithm minimize \( \sum_{j=1}^{m} (L - k_j) \) (the total unused length over \( m \) shelves)? If it does, give a proof. If it does not, give a counter example.

Please hand in your assignment electronically on Moodle.