CS340 Analysis of Algorithms
Fall 2025

Homework:	**4**	**Professor:**	**Elizabeth Dinella**
Due Date:	**10/6/25**	**E-mail:**	`edinella@cs.brynmawr.edu`
Office:	**Park 205**	**URL:**	`https://bmc-cs-340.github.io`

Reminder that project checkpoint 1 is also due next week, on Sunday 10/5.

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 ($1 \le i \le n$) 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 $0 \le p_i \le 1$, 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 ith box is the product of its sales probability and the sum of weight of all boxes above it and itself, that is, $C_i(S) = p_{s_i} \cdot (\sum_{j=1}^{i} w_{s_j})$. 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:

$$\begin{array}{lll} b_1: & w_1 = 300 & p_1 = 0.4 \\ b_2: & w_2 = 200 & p_2 = 0.35 \\ b_3: & w_3 = 500 & p_3 = 0.1 \\ b_4: & w_4 = 100 & p_4 = 0.15 \end{array}$$

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

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

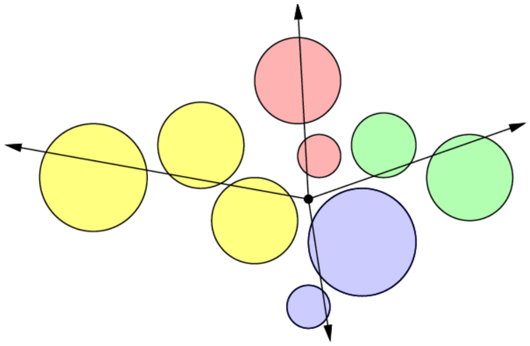
You may disregard any concerns of ceiling height, as well as dimensions and mechanical strengths of the boxes. In other words, assume that all permutations give stacking orders that are physically reasonable.

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

(b) Present a (short) counterexample to show that stacking the boxes in decreasing order of access probability p_i (from the top) 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. Prove your algorithm's correctness and derive its running time.

3. Full write-up. Suppose you are standing in the open surrounded by enemies, represented by circles of different sizes. You want to use your laser gun to destroy the enemies without moving from your current location. The enemies are stationary, your laser beam penetrates and you want to fire as few shots as possible.



The problem can be stated more formally as follows. Given a set C of n circles in the plane, each specified by its radius and the (x, y) coordinates of its center, compute the minimum number of rays from the origin that intersect every circle in C. Your goal is to find an efficient algorithm for this problem. This is a much simplified version of the more general "shooter-location problem", which has applications in graphics, robotics and image processing.

Assume that you have a function `intersect(r, c)` which determines whether an arbitrary ray `r` intersects an arbitrary circle `c` in $O(1)$ time. This is indeed true and easy enough to write, but not the interesting part of this problem. Assume also that there exists a ray that does not intersect any circles and you know where it is.

Please hand in your assignment on Gradescope.