

Dalhousie University
CSCI 3110 — Design and Analysis of Algorithms I
Fall 2025
Assignment 5

Distributed Tuesday, November 18 2025.

Due 11:59PM, Tuesday, November 25 2025.

Instructions:

1. Before starting to work on the assignment, make sure that you have read and understood policies regarding the assignments of this course, especially the policy regarding collaboration. <https://dal.brightspace.com/d2l/le/content/391731/Home?itemIdentifier=D2L.LE.Content.ContentObject.ModuleC0-5101579>
2. To submit via Crowdmark, upload a **separate** image/PDF file (multiple pages are allowed) to each question. You may resubmit as often as necessary until the due date. A good strategy is to create an initial submission days in advance after you solve some problems, and make resubmissions later. More detailed submission information for Crowdmark can be found at:
<https://crowdmark.com/help/completing-and-submitting-an-assessment/>
3. If you submit a joint assignment, use Crowdmark to create a group and add group members before making submissions. The “What Will Students See” section of <https://crowdmark.com/help/creating-a-group-assignment> shows the steps and screenshots of creating a group and adding members. Note that, “to avoid overwriting submissions”, Crowdmark recommends “only one group member submits all the work”.
4. We encourage you to typeset your solutions using LaTeX. However, you are free to use other software or submit scanned handwritten assignments as long as **they are legible**. We have the right to take marks off for illegible answers.

Questions:

1. [10 marks] The following statement is useful when arguing about the correctness of the binary algorithm of computing the greatest common divisor:

Let d be a positive integer, and u and v be positive odd integers with $u > v$. Then $d|u$ and $d|v$ if and only if $d|(u - v)/2$ and $d|v$.

Now prove this statement is true.

Note that you are NOT asked to use this statement to argue about the correctness of the binary algorithm. Rather, you are asked to prove the above statement itself.

Hint: Make sure to prove both the “if” and “only if” in this claim.

2. [10 marks] You are designing a load-planning module for a courier company. Each morning, the system must determine how to partition a day's deliveries into a fleet of trucks.

Crucially, the boxes are already pre-sorted based on the delivery route (e.g., by street address sequence). To allow the driver to deliver efficiently without searching through the truck, the boxes must be loaded in this exact pre-sorted sequence.

The Input: We are given a complete batch of n boxes, $B = \{b_1, b_2, \dots, b_n\}$, fixed in their delivery order. For each box b_i , we are given its weight w_i . We also have a fleet of identical trucks, where each truck has a maximum weight capacity of W . (Assume $w_i \leq W$ for all i).

Because this is a planning step performed on the full batch before physical loading begins, your algorithm has access to the weights of all items in the sequence.

The Objective: We need to partition the sequence B into groups, where each group constitutes one truckload. Our goal is to minimize a total “Inefficiency Metric” based on how much empty space is left in the trucks.

The Inefficiency Metric for packing a single truck with the contiguous sequence of boxes b_i, \dots, b_j is defined as:

$$M(i, j) = \left(W - \sum_{k=i}^j w_k - (j - i) \right)^3 \quad (1)$$

provided the base value (the term inside the parentheses) in Equation (1) is non-negative.

(To interpret Equation (1), consider that to pack boxes safely, a standard **separator** of weight 1 must be inserted between each adjacent box. Therefore, we start with capacity W , deduct the sum of the box weights, and then deduct $(j - i)$, which is the total weight of the separators. Thus, the equation represents the cube of the wasted weight capacity).

If the base value in Equation (1) is negative, then we set $M(i, j) = +\infty$, to indicate that the boxes exceed the truck’s limit.

Exception: If the calculated metric is positive but $j = n$ (meaning this truck carries the final box b_n), we override the formula and set $M(i, j) = 0$, as we do not penalize the final truck for being partially empty.

The total inefficiency for the day’s fleet is the accumulated sum of the metric values for all trucks used (recall that $M(i, j) = 0$ for the final truck, and any invalid truck renders the total inefficiency infinite).

The Greedy Approach: Consider a greedy algorithm that packs trucks sequentially. For the current truck, which starts at the next available box b_i , the algorithm selects

the largest possible index j (where $j \geq i$) such that the sequence of boxes b_i, \dots, b_j fits within the limit W . In other words, it packs as many consecutive boxes as possible into the current truck before moving to the next.

Give a specific counterexample (values for W and the list of weights w_i) where this greedy strategy fails to find the optimal solution (the minimum total inefficiency). Argue why the greedy choice is suboptimal in your example.

3. [10 marks] Refer to the problem statement, input definitions, and Inefficiency Metric $M(i, j)$ defined in the previous question.

Use dynamic programming to design an efficient algorithm to determine the minimum possible total inefficiency for the batch. You are only required to report the value of this minimum.