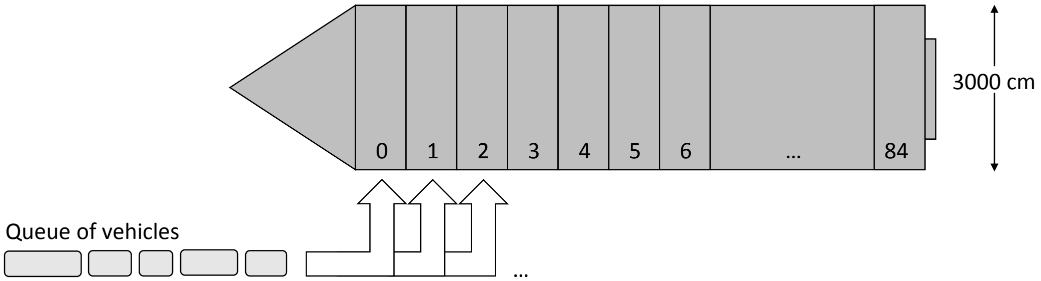
1 Summary

In this coursework you are to write a program that attempts to solve a problem involving loading vehicles on to a ferry. Your program will read in a problem instance from a file and will produce a solution using various simple optimisation techniques. You are also encouraged to alter various parameters to investigate the nature of the problem and your algorithm(s). Good programs will output information to the user during and after a run, and include graphical output.

2 Problem Specification

A large ferry is used to transport vehicles across an estuary. To load the ferry, the whole of one side of the boat is lowered, and each vehicle is instructed to drive into a particular lane. The ferry consists of 85 lanes in total (labelled 0 to 84) and each lane has a capacity (length) of 3,000 cm, as illustrated in the following diagram.



Currently the company that operates the ferry uses a very simple queueing system. Vehicles of varying lengths arrive at the port one-by-one and join a single queue. When the time comes to load the ferry, each vehicle is taken from the queue one at a time (in order) and is instructed to drive into *the lowest-indexed lane that currently has sufficient capacity for the vehicle*. If there are no lanes with sufficient capacity, the vehicle is sent to an overspill car park and is not allowed to board the ferry. Once this vehicle has been dealt with, the next vehicle in the queue is considered in the same way. This continues until all vehicles have been considered.

It is obviously in the ferry company's interest to minimise the amount of wasted lane-space in each ferry crossing. Hence our aim is to minimise the total length of the vehicles assigned to the overflow car park. This is equivalent to maximising the total length of all vehicles from the queue that are allowed to enter the ferry.

3 Problem Instances and Code

You have been supplied with example instance of this problem in **input.txt**. The first two lines of this file contain the length of each lane (3,000 cm) and the number of lanes (85), respectively. This is then followed by the (integer) lengths of 500 vehicles (in cm), in the order that they have joined the queue. Sizes of vehicles in this instance have been generated randomly according to the numbers in the following table. In each case, the size of vehicles belonging to each class have been generated uniform randomly in the given ranges.

Classification	Size Range (cm)	Quantity
Small car	$350 \leq \text{size} < 400$	100
Medium car	$400 \leq \text{size} < 450$	200
Large car	$450 \leq \text{size} < 500$	100
Van	$500 \leq \text{size} < 600$	70
Lorry	$600 \leq \text{size} \leq 2000$	30
		Sum = 500

You have also been supplied with some Python code that reads in this problem instance and assigns the vehicles to lanes using the process described above. In this code

- The variables `c` and `numlanes` are used to store the lane capacity and number of lanes respectively;
- The list `L` is used to hold the lengths and order of all vehicles in the queue
- The list of lists `S` is used to hold the vehicle lengths assigned to each lane in the ferry (the total of each lane cannot exceed `c`);
- The list `O` is used to store all vehicle lengths that have been assigned to the overflow car park.
- The function `getFirstLane` is used to determine the lane for each vehicle. If no lane is suitable for a vehicle due to insufficient capacity, the function returns `-1`.

Download the above files onto your computer (putting them into the same folder) and get them running. Ensure that you understand what each line of code is doing.

4 Task 1

As noted, the current strategy for choosing a lane is to use the lowest-indexed lane that has sufficient capacity for the vehicle. However, it has been suggested that some other strategies might be more appropriate. These include, for each vehicle:

1. Choosing the emptiest lane with sufficient capacity for the vehicle;
2. Choosing the fullest lane with sufficient capacity for the vehicle;
3. Choosing any random lane with sufficient capacity for the vehicle;

where "emptiest" and 'fullest" are determined using the total length of all vehicles currently assigned to a lane.

Using the function `getFirstLane` in the supplied code as a guide, program the above rules (and perhaps others) and investigate their effects on the quality of your solution.

- At the start of a run, you may wish to ask the user which of the above rules they want to use to load the ferry.

- In addition to text output, you may also wish to use various graphical outputs (charts) to illustrate your solutions. These can also be included in your report, if desired.

5 Task 2

The ferry company has decided to invest some money at the port to try and further increase efficiencies. Suggestions include:

1. Using five separate queues, one for each vehicle type (as defined in the table earlier). The ferry is then loaded using the lorry queue first, then the vans queue, then large cars, then medium cars, and then small cars.
2. Keeping the current system of one queue but, in each step, taking the next k vehicles in the queue, reordering these k vehicles from longest to shortest, and then inserting each of these on to the ferry in turn.

Using your best performing rule from Part 1, investigate the efficiency of these proposed strategies, and perhaps others, using **input.txt**.

- The parameter k needs to be decided upon by the user and can range from 1 to n, where n is the total number of vehicles in the queue at the start of the boarding process. You may wish to experiment with different values here.
- As before, you may also wish to use various graphical outputs (charts) to illustrate your solutions.

6 Task 3

One flaw in the tests conducted so far is that they only apply to the single problem instance given in **input.txt**. Indeed, if the vehicles had arrived in a different sequence, or were of different lengths then different conclusions might be drawn.

In order to convince the ferry operator that your proposed methods will bring more efficiencies, you decide that wider tests are required. This should involve randomly generating multiple problem instances (using the same classifications and quantities as the table earlier), re-running your methods, and documenting your results.

You may also want to widen your study to look at what might happen if a boat with a different number of lanes and/or different lane capacities are used. Conduct some investigations into this.

7 Extra Information

- You will have noticed that some of these tasks are open ended. You are therefore free to choose your own directions for these parts.
- Extra marks will be allocated to those who write neat, elegant, efficient code that is appropriately commented. It is desirable for the program to give the user prompts, allowing them to choose between different run options.
- Marks are also allocated to reports that are written accurately and succinctly, with appropriate typesetting. You may also wish to use various figures and tables in your report.
- It is also acceptable to include small pieces of your Python code in your report if it helps to explain something. However, bear in mind that you are also required to submit your code in a separate file.

8 Assessment and Submission

Submission for this coursework is in two parts.

Part 1 For the first part of your submission, you are asked to email **email your lecturer** a working version of your Python program in a single **.py** file.

Marks will be allocated subject to the above tasks being programmed correctly, neatly and efficiently. Note that it is usually preferable to have a less-advanced program that runs correctly rather than a more advanced program that does not run or that has bugs in it.

Part 2 For the second part of your submission, you are asked to submit a two-page scientific report that summarises your investigations. Good marks will be reserved for reports that are written accurately and succinctly, with appropriate typesetting. A suggested structure is to have sections headed Aims, Methods, Analysis, and Conclusion. You may also wish to use various figures and tables in your report. In your investigations, statistics should be presented to back your assertions.