Project 1 - CMPE Truck Simulator
CmpE 250, Data Structures and Algorithms, Fall 2024

SA: Işıl Su Karakuzu
TAs: Kutay Altıntaş, Beyza İrem Urhan

Due: 07/11/2024, 23:55 Strict

1 Introduction

You and your friends have decided to take a break from computer engineering and start a truck company. Your task is to manage a fleet of trucks with varying capacities, ensuring that they are assigned to jobs efficiently and effectively.

2 Organization

Your truck fleet consists of trucks with different capacities. Each truck is assigned to a parking lot based on its capacity constraint.

Example: Truck 1 (60 unit capacity, 10 units of load) and Truck 2 (50 unit capacity, 0 units of load) will both be placed in the parking lot with a capacity constraint of 50.

Each parking lot has two sections: a waiting section and a ready section. Trucks must be in the ready section to move to a new parking lot. More details about this structure are provided below.

2.1 Trucks

- Each truck has a unique integer ID and a maximum capacity.
- Trucks cannot be loaded beyond their maximum capacity.

2.2 Parking Lots

- Each parking lot has a unique capacity constraint. In a parking lot with a specific capacity constraint, only trucks with that exact amount of remaining capacity can be allocated there.

  Example: Truck 1 (30 unit capacity, 0 units of load) and Truck 2 (60 unit capacity, 30 units of load) will both be placed in the parking lot with a capacity constraint of 30.

- The parking lot will consist of two sections: waiting and ready. Each parking lot will also have a truck limit, which indicates the maximum number of trucks it can hold in both sections combined.

  Example: Capacity Constraint: 30 Truck limit: 3
  Waiting section: Truck 1 (30 unit capacity, 0 units of load)
  Ready section: Truck 2 (60 unit capacity, 30 units of load), Truck 3 (100 unit capacity, 70 units of load)
  Total trucks: 3 (limit reached)

- When a truck enters a parking lot, it is initially placed in the waiting section. If the intended lot is full, the truck moves to the largest available lot with a smaller capacity until one is found. If no suitable lot exists, the truck is not added.

  Example:
  Truck with ID 1 and maximum capacity 50 is added.
  Parking Lot with capacity constraint 50: Truck limit full (check next smaller).
  Parking Lot with capacity constraint 40: Truck limit full (check next smaller).
  Parking Lot with capacity constraint 30: Not full. Truck is added to the waiting section (**This truck can load up to 30 units despite having a maximum capacity of 50**).

- When a ready command is issued, the earliest waiting truck (the truck that arrived to waiting section first) will be moved to the ready section of that parking lot. If no trucks are waiting, smaller capacity lots are checked from larger to smaller until a waiting truck is found. If none are found, no truck will move.

Example: In Parking Lot with capacity constraint 50, if there are no waiting trucks, check Parking Lot with capacity constraint 40, then Parking Lot with capacity constraint 30. If Parking Lot with capacity constraint 30 has a waiting truck, move it to the ready section of Parking Lot with capacity constraint 30. If all waiting sections of checked lots are empty, no truck will move.

- If a parking lot receives a unit of load, it will be assigned to the earliest truck in the ready section. If the load exceeds the remaining capacity of the earliest truck, the next truck will be selected, and the process will continue until all the load is distributed. If any load still remains, it will be transferred to the next parking lot with the smallest capacity greater than the current parking lot's capacity. This transfer will not use any of your trucks, the provider will transfer the load.

  Example: Received units of load: 100

  Parking Lot 1: Capacity constraint: 30
  Waiting: Truck 1 (30 unit capacity, 0 units of load)
  Ready: Truck 2 (60 unit capacity, 30 units of load), Truck 3 (100 unit capacity, 70 units of load)

  Parking Lot 2: Capacity constraint: 60
  Waiting: Truck 4 (70 unit capacity, 10 units of load)
  Ready: Empty

  Parking Lot 3: Capacity constraint: 80
  Waiting: Empty
  Ready: Truck 5 (150 unit capacity, 70 units of load)

  Load Distribution Process:

  Truck 2 (60 unit capacity) takes 30 units, bringing its load to 60 units (full).
  Truck 3 (100 unit capacity) also takes 30 units, bringing its load to 100 units (full).
  Remaining load: 40 units.

Next, we check parking lots with a capacity constraint greater than 30, from smallest to largest:

Parking Lot 2 (capacity constraint: 60): No ready trucks available.
Parking Lot 3 (capacity constraint: 80): Truck 5 is ready and will take the remaining 40 units, increasing its load to 110 units.

- No matter what happens, trucks in the waiting section will not be used for transfers until they are moved to the ready section of the parking lot.
- When trucks receive their load, there are two possibilities:

  1. The truck is fully loaded to its maximum capacity, with no space left. In this case, the truck will automatically go unload all of its load and then return to its new parking lot. The starting and finishing parking lots may be different.

     Example:
     Truck 1: Capacity: 70, Load: 10
     Now in parking lot: Capacity constraint: 60
     Receives 60 units (total 70 units of load, full)
     Unloads all 70 units and moves to parking lot with capacity constraint: 70

  2. The truck is loaded but not at maximum capacity. In this case, it will head to the corresponding parking lot that matches its remaining capacity constraint.

     Example:
     Truck 1: Capacity: 70, Load: 10
     Now in parking lot: Capacity constraint: 60
     Receives 30 units (total 40 units of load, not full)
     Moves to parking lot with capacity constraint: 30

2.3 Management of the Fleet

- Your company will manage parking lots by adding and deleting them, along with adding, loading, and moving trucks between lots. You are expected to

implement an efficient structure to handle these tasks and ensure the fleet is managed effectively.

3 I/O Files

The input file consists of a series of actions, and for each action, a specific type of logging or response is expected. The details for each action and the corresponding outputs are provided below. In each action, if there is a corresponding output, it should be written to the **output.txt** file. Here's a sample structure for the input file:

```
create_parking_lot 50 15
add_truck 100
ready 100
load 100 50
delete_parking_lot 50
count 50
```

For more details about the actions in the input file, please refer to the actions section.

4 Actions

4.1 Creating New Parking Lot

You should create a parking lot with the given capacity constraint and truck limit, including its waiting and ready sections. Initially, both sections will have 0 trucks. If a parking lot with that capacity constraint already exists, no action will be taken.

Structure: **create_parking_lot <capacity_constraint> <truck_limit>**
Example: **create_parking_lot 50 15**

Explanation: Trucks with 50 units of remaining capacity will be assigned to this lot. Truck 1 (60 capacity 60, 10 load) Truck 2 (50 Capacity, 0 load) will be placed in this lot. The maximum truck count in this lot cannot exceed 15. For instance, if there are 10 trucks in the waiting section and 5 in the ready section of this lot with capacity 50, the lot is at full capacity.

Output: No output will be created for this action.

4.2 Deleting Parking Lot

You should delete the parking lot with the specified capacity constraint, including its waiting and ready sections and any trucks inside them. If a parking lot with that capacity constraint does not exist, no action will be taken.

Structure: **delete_parking_lot <capacity_constraint>**
Example: **delete_parking_lot 50**

Explanation: The parking lot with a capacity constraint of 50 will be deleted, along with both its waiting and ready sections. Any trucks inside these sections will also be removed. If any trucks have a load, that load will be deleted as well.

Output: No output will be created for this action.

4.3 Adding a Truck

You should add the truck with the specified capacity to its corresponding parking lot. Every added truck will go into the waiting section and remain there unless a ready command is issued.

If a parking lot with the specified capacity constraint does not exist or is full, the truck should be placed in the largest parking lot with a capacity smaller than its own. This process will continue until an available parking lot is found. If no such parking lot exists, or if all smaller parking lots are full, the truck will not be added, and no action will be taken. The truck can only use the capacity constraint of the parking lot in which it is placed.

Structure: **add_truck <truck_id> <capacity>**
Example: **add_truck 1 100**

Explanation: A truck with ID 1 and a total capacity of 100 will attempt to be added to the waiting section of its corresponding parking lot, 100. If a parking lot with a capacity constraint of 100 doesn't exist or is full, the truck will be placed in the largest available parking lot with a smaller capacity. If no such parking lot exists or all are full, no action will be taken. All trucks added using this method will start with 0 load initially.

Output: <**capacity**> or **-1**

You should write the capacity of the parking lot where the truck is placed to a line in the `output.txt` file. If the truck could not be placed, write **-1**.

4.4 Ready Command

When this command is issued with a parking lot capacity constraint, the earliest arrived truck in the waiting section of that parking lot will be marked as ready and moved to the ready section. If there are no trucks in the waiting section, the command will search for the earliest waiting truck in the smallest available parking lot with a larger capacity. This process will continue until a waiting truck is found. If no trucks are found in any larger parking lots, no action will be taken.

Structure: **ready** <**capacity**>
Example: **ready 50**

Explanation: When the **ready 50** command is issued, it will check the waiting section of the parking lot with a capacity constraint of 50. If there is a truck in the waiting section, the earliest waiting truck will be marked as ready and moved to the ready section. If the waiting section of the 50 capacity parking lot is empty, the command will search for the earliest waiting truck in the smallest available parking lot with a capacity larger than 50. This process will continue until a waiting truck is found. If no waiting trucks are found in any larger parking lots, no action will be taken.

Output: <**truck_id**> <**capacity**> or **-1**

You should print the ID of the truck that has been moved into the ready section, along with its corresponding parking lot capacity, on the same line, separated by a whitespace. If no truck is found to be added to the ready section, print **-1**.

4.5 Recieving a Load

In this action, a provider gives a load to a specific parking lot. This load may be greater or lesser than the capacity of the trucks in that parking lot. Only the trucks

in the ready section can accept loads. The load will be given to the earliest arrived truck in the ready section. If there is remaining load after the first truck is loaded, the next earliest arrived truck in the ready section will take the leftover load, continuing this process until all the load is distributed.

If all trucks in the ready section have been loaded and there is still a portion of the load remaining, the provider will transfer the leftover load to the smallest capacity parking lot that is larger than the current parking lot. The load will then be distributed among the trucks in the ready section of this new parking lot. If there are no ready trucks available or no such parking lot exists, the remaining load will go to waste, and no further action will be taken.

Request:

Structure: **load <capacity> <load_amount>**
Example: **load 50 200**

Explanation: The command signifies a request to distribute a total load of 200 units among the trucks in the ready section of the parking lot with a capacity constraint of 50. Assuming there are three trucks in the ready section, each with a remaining capacity of 50, the command will first allocate 50 units to the first truck, leaving 150 units of load remaining. Next, the second truck will receive another 50 units, reducing the load to 100 units. Then, the third truck will also be assigned 50 units, bringing the remaining load down to 50 units. Since all three trucks have reached their maximum capacity, and there is still 50 units of load left, the command will then look for the smallest available parking lot with a capacity greater than 50 to continue distributing the remaining load. If no such parking lot exists or they are all full, the remaining load will go to waste.

Output: **<truck_id1> <capacity_constraint_of_new_load>(if not placed** -1**) - <truck_id2> <capacity_constraint_of_new_load>** ...

Example: load 50 120

Parking Lot 1: Capacity constraint: 50
Waiting: Truck 1 (50 unit capacity, 0 units of load)
Ready: Truck 2 (60 unit capacity, 10 units of load), Truck 3 (100 unit capacity, 50 units of load)

Parking Lot 2: Capacity constraint: 60
Waiting: Truck 4 (70 unit capacity, 10 units of load)
Ready: Empty

Parking Lot 3: Capacity constraint: 80
Waiting: Empty
Ready: Truck 5 (150 unit capacity, 70 units of load)

Load Distribution Process:

Truck 2 (60 unit capacity) takes 50 units, bringing its load to 60 units (full).
Truck 3 (100 unit capacity) also takes 50 units, bringing its load to 100 units (full).
Remaining load: 20 units.

Next, we check parking lots with a capacity constraint greater than 50, from smallest to largest:

Parking Lot 2 (capacity constraint: 60): No ready trucks available.
Parking Lot 3 (capacity constraint: 80): Truck 5 is ready and will take the remaining 20 units, increasing its load to 90 units.

In conclusion:

- Truck 2 is full, it will unload all its load and move to the parking lot with a 60 capacity constraint.
- Truck 3 is also full; it will unload all its load and move to the parking lot with a 100 capacity constraint. Since there is no such parking lot, it will move to the largest available lot with a smaller capacity, which is 80. Now, Truck 3 can only take 80 units of load in this parking lot, even though its maximum capacity is 100.
- Truck 5 is not full (it has 90 load), it will move to the parking lot with a 60 capacity constraint (since its remaining capacity is 150-90).

Output: **2 60 - 3 80 - 5 60**

4.6 Requesting Truck Count for Trucks with Capacity Larger than Given Capacity Constraint

This action requests the truck count for trucks in parking lots with a capacity constraint **greater than** the given capacity constraint, considering both the ready and waiting sections of those lots.

Structure: **count <capacity>**
Example: **count 50**

Explanation: The command requests the total number of trucks in parking lots with a capacity constraint **greater than** 50 units (Therefore, 50 itself is not included.). It identifies all relevant parking lots with a capacity constraint greater than 50 units and counts the trucks in both the ready and waiting sections of those lots.

Output: <**truck_count**>

You should write the truck count in parking lots with a larger capacity constraint to a line in the `output.txt` file.

5 Submission

You will be submitting a zip file containing your code via Moodle. Your code must be able to run correctly with these commands:

javac *.java
java Main <input_file> <output_file>

6 Grading and Constraints

6.1 Grading Specification

The inputs consist of five types, each with specific operations and corresponding point values for grading:

Type 1: Create Lot and Add Truck Operations (30 points)
Implement methods to create and add lots to the system and add trucks while keeping in mind the truck limits of each lot.

Type 2: Create Lot, Add Truck, and Ready Operations (20 points)
Extend Type 1 by adding the ability to mark trucks as ready for use.

Type 3: Create Lot, Add Truck, Ready, and Load Operations (30 points)
Include loading functionality for ready trucks in specified lots, ensuring the capacity limits of trucks are respected.

Type 4: Create Lot, Add Truck, Ready, Load, and Delete Lot Operations (10 points)
Allow lots to be deleted from the system without disrupting fleet operations. Trucks also need to be deleted from the system.

Type 5: Create Lot, Add Truck, Ready, Load, Delete Lot, and Count Operations (10 points)
Combine all previous operations and implement a counting feature to track the truck count for trucks in parking lots with a capacity constraint **greater than** the given capacity constraint.

The point distribution for small and large cases is as follows:

Type	Small Case (%60)	Large Case (%40)
1	18	12
2	12	8
3	18	12
4	6	4
5	6	4

Table 1: Grade Distribution for Small and Large Cases

6.2 Constraint Specification

Constraint specifications for each input type are given below.

Input Type	Type 1		Type 2		Type 3		Type 4		Type 5	
	Small	Large	Small	Large	Small	Large	Small	Large	Small	Large
max_lot_id	50	200000	100	500000	100	500000	100	500000	100	500000
num_create_lot	50	200000	50	200000	20	100000	20	50000	20	100000
num_delete_lot	0	0	0	0	0	0	10	30000	3	1000
num_add_truck	200	500000	200	500000	100	500000	100	10000	100	200000
num_ready_truck	0	0	200	500000	300	500000	200	500000	200	200000
num_load	0	0	0	0	500	500000	500	500000	500	200000
num_count	0	0	0	0	0	0	0	0	100	500000
truck_count_limit	3	5	3	5	3	100000	3	3	3	5
truck_capacity	50	200000	100	500000	100	500000	100	500000	100	500000
load_limit	100	1000000	400	500000	200	500000000	200	5000000	200	500000

Table 2: Maximum Values of Different Types (Small and Large Versions)

The following constraints apply to all types and versions in the table above:

- **max_lot_id:** The maximum allowable ID for lots, which is also the capacity constraint of the lot.
- **num_create_lot:** The maximum number of **create_parking_lot** operations in an input file.
- **num_delete_lot:** The maximum number of **delete_parking_lot** operations in an input file.
- **num_add_truck:** The maximum number of **add_truck** operations in an input file.
- **num_ready_truck:** The maximum number of **ready** operations in an input file.
- **num_load:** The maximum number of **load** operations in an input file.
- **num_count:** The maximum number of **count** operations in an input file.
- **max_truck_limit:** The maximum number of trucks allowed in a lot. This does not guarantee that all lots have this limit.
- **max_truck_capacity:** The maximum load capacity of a truck, defining how much each truck can carry. This does not ensure all trucks have this capacity.

- **max_load:** The maximum load that can be given to a lot in a load operation. This does not guarantee that all load operations have this much load.

6.3 Time Constraints

Each test case has a time limit of 25 seconds. Our code completes each test case in under 10 seconds. These durations were measured on an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz. Ensure that your program's runtime does not exceed 25 seconds, as the automatic grader will terminate execution after this limit.

7 Warnings

- **Any structure other than ArrayList is forbidden; you cannot use other Java.util classes.**
- All source codes are checked automatically for similarity with other submissions and exercises from previous years. Make sure you write and submit your own code. Any sign of cheating will be penalized by at least -100 points at first attempt and disciplinary action in case of recurrence.
- Make sure you document your code with necessary inline comments and use meaningful variable names. Do not over-comment, or make your variable names unnecessarily long. This is very important for partial grading.
- Make sure that the white-spaces in your output is correct. You can disregard the ones at the end of the line.
- Please use the discussion forum at moodle for your questions, and check if it is already answered before asking.