Norges Teknisk-Naturvitenskapelige Universitet

TPK4186 - Advanced Tools for Performance Engineering Spring 2023

Assignment 1: Container Ships

Prepared by Antoine Rauzy

Contents

1 Introduction

1.1 Presentation of the Problem

You are asked by the shipping company Trondheim International Shipping (TIS) to design a program to help them with loading and unloading ship containers and to do some book-keeping of containers arriving and departing from the Trondheim harbor. Here follows some information about the containers, the ships and the harbor.

Containers We shall consider two types of containers: 20 feet long containers and 40 feet long ones. As a realistic simplification of the reality, we shall assume that:

- A 20 feet long container weights 2 tons and can be loaded with up to 20 tons of freight.
- A 40 feet long container weights 4 tons and can be loaded with up to 22 tons of freight.

Each container is uniquely identified with a code. We need also to keep for each container, its size and its (loaded) weight, and may be some additional information.

Container Ships Again to make things simple, we shall see that container ships as 3 dimensional boxes. Each cell of the box, called a bay, can contain a 20 feet container. The number of bays in thus $L \times W \times H$ box, where L length of the ship, W its width and H its height, all measured in number of 20 feet containers.

The capacity of container ship is actually measured twenty-foot equivalent units (TEU). Typically $L = 23$, $W = 22$ and $H = 18$. See Figure 1 for an illustration.

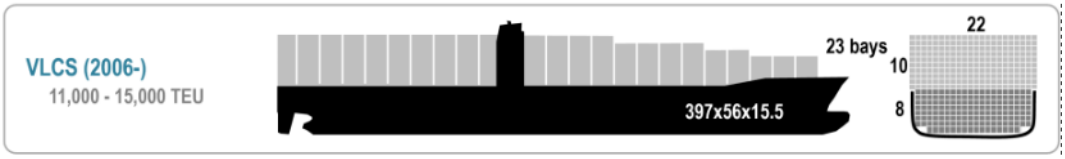

Figure 1: A container ship

Harbor

1.2 Requirements

The objective of this assignment is to show your Pythonic skills.

Here follows a number of requirements.

1. You must provide your program together with a small document explaining how it is organized, what it is doing (which functionalities are implemented) and reporting experiments you have performed with it.

2. The assignment can be made either individually or in group (maximum 3 persons, preferably 2).

3. Assuming you named your group $Python's fan$, all the files you deliver must be included in a zip archive named:

   `TPK4186 - 2023 - Assignment 1 - Python's fan.zip`

   The deliverable of the assignment is this zip archive. Please recall your names in addition to the group name both in the header of the program and in the accompanying document.

4. The quality of a program is judged along three criteria: its completeness, its correctness and its maintainability:

   – A program is complete if it provides all functionalities demanded by the client. Some functionalities are however more important than other. You must first concentrate on the main functionalities, then develop the "nice-to-have" ones.

   – A program is correct if it is bug free. To ensure that your program is correct, you must test it extensively. Design tests before writing the first line of code. There is no such a thing than a program or a functionality that works "most of the time". Either it works, or not. If you are not able to make a functionality work, do not deliver it.

   – A program is maintainable if it is well presented, if the identifiers are significant, and so on. But before all, a program is maintainable if it well organized and as modular as possible. Separate the concerns.

2 Tasks

For this assignment you are asked to design several data structures and to implement quite a few functions. Each of these functions must be thoroughly tested.

The assignment consists in the following tasks.

2.1 Containers

We shall first implement functions to manage containers.

Task 1. Design a data structure to encode containers and implement associated the set and get functions.

Task 2. Design a data structure to encode a set of containers and implement the associated management functions (add a container, remove a container, look for a container...).

Task 3. Design a function that generates a container "at random", by choosing a length, a load and code for the container. This function may be parametric. Design a function (using the previous one) that generates at random a set of containers.

Task 4. Design a function that prints out a set of containers into a file and another function that loads a set of containers from a file. The file must contain one line per container. Information relative to one container (identifier, length, weight empty, weight loaded...) must be separated with tabulations. This format for file is called TSV (tabulation separated values), or sometimes CSV (comma separated values).

2.2 Ships

Task 5. Design a data structure to encode container ships implement the associated management functions: look for a container in the ship, look for a place where a container can be loaded, load a container into the ship, remove a container from the ship.... Note that it is not possible to load a 40 feet container onto a single 20 feet container (there should no holes).

Task 6. Design a function that prints out the load of a ship into a file and function that loads the load of a ship from a file. Propose a TSV format for these files.

Task 7. Design a function that loads a ship container by container from a set of containers and creates the corresponding ordered list of containers. Design a function that unload a ship, container by container and creates the corresponding ordered list of containers.

Hint: Note that because of the constraint that there must be no hole, it may be not possible to load a given container into a partially loaded ship. In this case, a possible solution consists in removing the last container that was loaded, to try to load the new one and put back the container that was removed.

Note also that it may be impossible to load all of the containers of the given set, even using the above "trick".

Task 8. For stability reasons, heady containers must be placed below light ones. Design a new loading function that takes into account this constraint, i.e. that piles containers in decreasing weight order.

Task 9. Design functions that calculate:

- The total weight of containers loaded in a ship.
- The total weight of containers loaded on starboard and on portside of ship.
- The total weight of containers loaded the first, middle and last section of a ship (from the bow to the stern).

The load of ship is balanced if:

- Containers are piled in decreasing weight order.
- The weight on starboard does not exceed the weight on port side by more that $x\%$, e.g. $x = 5$.
- The weight on a section of the ship does not exceed the weight on another section by more that $y\%$, e.g. $y = 10$.

Using the above functions, design a function that tells whether the load of a ship is balanced.

Task 10. Design a new loading function that takes into account the stability constraints.

2.3 Docks

We shall assume now that 4 cranes are available to load and unload ships.

Indeed, two cranes cannot operate in the same part of the ship, due to the danger of collision. The ship is thus divided in four sections, from the bow to the stern. The additional rule is that the two cranes cannote load or unload containers in adjacent bays.

Task 11. Assuming that it takes about 4 minutes to load or unload a container with a crane. Design a function that calculates how much time it takes to load or unload the ship with 1 crane.

Task 12. Same questions with 4 cranes. What do you observe?