Ahmed Absal Mohamed (S2101349) (21055511)

Bigcon construction

UFCFW4-30-2 - Design and Analysis of Data Structures and Algorithms Coursework

12 Dec 2022

# Introduction

BigCon, a large construction company has sites in 4 un-inhabitant islands (A, B, C and D). Currently the company is constructing housing facilities at these islands. It owns only one dhoani to carry consumables from supplier islands to construction islands.

The purpose of this coursework is to assist the company you are required to design and implement an appropriate software to undertake the following tasks while assessing the best algorithms and data structures that should be implemented in terms of space and time complexity:

- Taking goods for consumption.

- Accept and add items into dhoani and island warehouses.

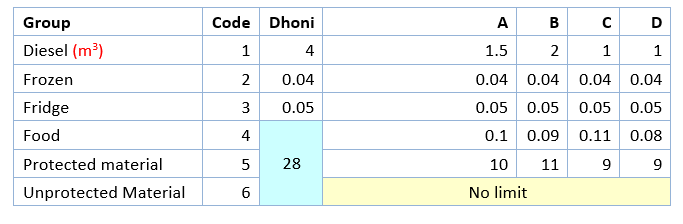
- Swap items between dhoani and island.

- Reject quantities of item beyond capacity.

# Identify and justify the choice of data structures and algorithms

## Assessing the current scenario

BigCon, a large construction company has sites in 4 un-inhabitant islands (A, B, C and D). Currently the company is constructing housing facilities at these islands. It owns only one dhoani to carry consumables from supplier islands to construction islands.

­­

Each site storages for the types mentioned above with the maximum space available defined.

Dhoani travels between supplier islands at two ends visiting construction islands in order from A to B to C to D or D to C to B to A. Dhoni will not stop at an island where the capacity is full (for all product groups).

The following figure gives information about the distance between the islands.



**Each site has a:**

* **#1** tank for storing diesel - **Code 1**
* **#2** deep freezer for storing frozen food such as meat - **Code 2**
* **#3** fridge for keeping vegetables, fruit and other food items requiring low temperature **- Code 3**
* **#4** room with a raised floor to store other food items such as grains, spices, etc **- Code 4**
* **#5** room for storing construction materials and other consumables such as cement, chemical, detergents, etc. **- Code 5**
* **#6** Some items such as steel, brick, river sand and aggregate are kept outside warehouse. They are grouped as unprotected material. **- Code 6**

**Rules**

* #7 Items are grouped based on its storage nature
* #8 A construction site will not accept goods beyond storage capacity.
* #9 A site may opt to remove some existing items to make space for other items (with the same code) and may unload some of their stock into dhoani.

**Looking at the rules that dhoani has to follow and the requirements that has to be met:**

* \***1** As per the company safety policy, dhoani is not allowed to exceed a limit of 30,000kg of goods.
* \***2** Dhoani has separate storage facility/area for storing items with group code from 1-3. Items with group code 4-6 has a common storage (see the table above).
* \***3** *For the sake of calculating the load, volume of diesel are converted into weight using the conversion rate `m3 of diesel  = 832kg of diesel*
* \***4** Dhoani captain is advised to occupy the maximum capacity permitted as far as possible. Note that some goods may remain in dhoani unloaded when it arrives at supplier’s island for loading goods. Therefore when loading new stock into dhoani, existing occupancy has to be considered.
* \**~~5* Dhoani staff accommodations are arranged at supplier’s islands. (not important)~~
* \***6** Dhoani travels between supplier islands at two ends visiting construction islands in order from A to B to C to D or D to C to B to A.
* \***7** Dhoni will not stop at an island where the capacity is full (for all product groups).
* \***8** Goods are loaded into the boat during night time.
* \***9** Dhoani travels only during the day time (from 6am to 6pm).
* \***10** Assuming that the unloading time at islands is zero, journey from one end to the other end (one supplier island to another supplier island) takes 12 hours if dhoani goes at its normal and average speed (25km/hour).
* \***11** When dhoani reaches an island, captain needs to know how many minutes of travelling is required to reach different islands ahead (islands yet to be reached during the day).
* \***12** Dhoani cannot accept any quantity exceeding their load capacity of any group of items.

Taking in to considerations these rules, policies, and contraints, a software capable of the following functions has to be implemented.

**Main Functions**

* **A** Taking goods for consumption.
* **B** Accept and add items into dhoani and island warehouses.
* **C** Swap items between dhoani and island.
* **D** Reject quantities of item beyond capacity.

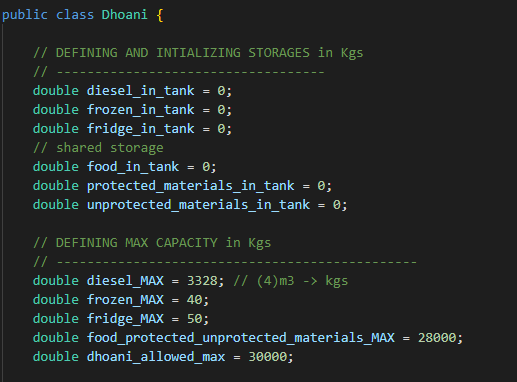
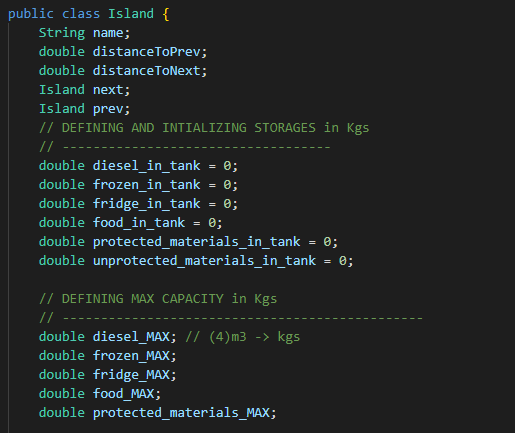
## The Data Structure to be used

When looking at the system in a birds eye view, it is evident that 2 types of objects with similar data structures are seen.

* **Dhoani x 1**
* **Supplier\_Island x 2 , Un-Inhabited\_Island x2**

All of these objects have quiet similar storage capabilities with different max capacities. The dhoani also has a shared storage space area for food, protected materials, and unprotected materials.

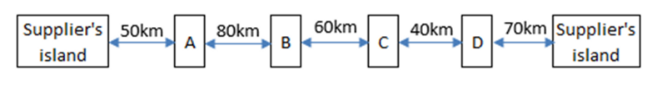
The software in this assignment is and will be implemented in Java. Since, the language is object oriented focused, the best implementation for the dhoani and islands would be to make a separate class for dhoani and island with variables defined as follows.

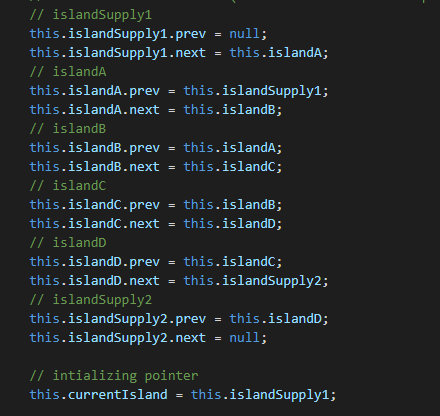
One thing to note that MAX values of an item storage will be defined, for storages with infinite MAX capacities, an extremely high value is assigned. For supplier islands, storages are initially filled with extremely high values that will not run out considering the dhoani’s max capacity.

The reason that the dhoani class had to be separated from the island class is due to its shared storage space. From this class layout, 1 dhoani object, and 6 island objects (Supplier Island 1, Island A, Island B, Island C, Island D, Supplier Island 2) can be made.

The dhoani can travel back and forth from one island to another without skipping the adjacent island.



In this scenario, the best data structure to chain these islands together would be to implement a **double linked list**. This is because its easy to traverse from one node to another in doubly linked lists, both forwards and backwards. A doubly linked list also ensures that the dhoani can traverse from one island (node) to another. It is also important to note that the scenario has already defined the number of islands and the structure, hence it is not important to think about the insertion and deletion of nodes.



## Algorithms to be used

In order to define algorithms, its important to first define the main tasks

1. Consumption from island
2. Add Items to Dhoani From Island
3. Add Items to Dhoani From Island
4. Travel From one Island to Another

After defining the main tasks properly, the following algorithms were created keeping in the constraints mentioned in section 2.1. of this documentation

1. Consuming from island
   1. Check if it is night time
      1. Check if current time is in between 6pm to 6am
      2. Return true or false value
   2. Check if the requestion amount to be consumed is available
      1. Get the requested amount
      2. Compare with the amount in tank
      3. Return true or false value
   3. If step 1.1, 1.2 are true Remove item from island tank
2. Adding Items to Dhoani From Island
   1. Check if it is night time
      1. Check if current time is in between 6pm to 6am
      2. Return true or false value
   2. Check if there is enough space available from dhoani
      1. Get the requested amount
      2. Compare it with the max storage amount with the requested amount + already existing amount
      3. Compare the requested amount and see if dhoani total capacity of 30,000kg is not exceeding
      4. Return true or false value
   3. Check if the requested amount is available from island
      1. Get the requested amount
      2. Compare the requested amount with existing amount to see if available
      3. Return true or false value
   4. If all of the previos conditions were true, swap items from island to dhoani
      1. Remove requested amount from island
      2. Add requested amount to dhoani
3. Adding Items to Island From Dhoani
   1. Check if it is night time
      1. Check if current time is in between 6pm to 6am
      2. Return true or false value
   2. Check if there is enough space available from island
      1. Get the requested amount
      2. Compare it with the max storage amount with the requested amount + already existing amount
      3. Return true or false value
   3. Check if the requested amount is available from dhoani
      1. Get the requested amount
      2. Compare the requested amount with existing amount to see if available
      3. Return true or false value
   4. If all of the previos conditions were true, swap items from dhoani to island
      1. Remove requested amount from dhoani
      2. Add requested amount to island
4. Travel (applicable when traveling to east or west)
   1. Check if its is day time
      1. Check if current time is inbetween 6am to 6pm
      2. Return true or false value
   2. Check for island edge case
      1. Check If current\_island.next == null
      2. Return true or false value
   3. Check if the next island with space available can be travelled
      1. While current\_island.next is not null
         1. Check if next island has space available
            1. If not check next next island
      2. The while loop is broken when an island with space is found or when an edge island is reached
      3. Return the island that can be traveled to with total time taken
   4. Check if the island returned from 4.3 can be traveled during the day time
      1. Get total time taken from 4.3.4 and add it with the current time
      2. Check if it exceeds the curfiew when added with the current time
      3. Return true or false
   5. When all of the above steps are true, move to next island
      1. Move to next island
      2. Calculate the time taken to reached the island with distance and speed given
      3. Add the time taken to current time
   6. If 4.5 is successful then check and display the furthest island that the dhoani captian can travel to during day time
      1. While the next island is not null (edge case check)
         1. Travel to next island
         2. Get time taken
         3. Add time to current time and check if curfew is passed
      2. The loop is broken once the curfew is passed or edge island is reached
      3. The furthest island that can be traveled during day time is returned to be displayed

## Analyzing space and time complexity

**TIME COMPLEXITY**

The length of the input determines how long it takes an algorithm to run, which is known as time complexity. It calculates how long it takes for each algorithm's code statement to run.

Looking at the algorithms defined in section 2.3 of this documentation, function 1, 2, & 3 accomplishes its task by only using if else statements without the need for need for any while loops or recursive functions. Hence, these three function displays a constant big oh with a big oh notation of **O(1)**

**Time**

**Input**

**O(1)**

For the 4th function we can see that the algorithm has two while loop in 4.3.1 and 4.6.1. The while loops are independent of each other and not nested. More over while there is a chance for both while loops to be not executed, which will result in a **constant time** complexity for the best case scenario.

If we look at the 4th function travel function, two of the while loops will depend on the number of islands (6) in the doubly linked list.

Excluding the current island, the maximum number of island will be n-1 which is equal 5. Hence, the big O notation of the 4th function would also be constant **O(1).**

**SPACE COMPLEXITY**

The amount of memory needed to solve a specific instance of the computational issue as a function of the input characteristics is known as the space complexity of an algorithm or computer program. It is the memory that an algorithm needs to run completely.

The main data structure used for this is a doubly linked lists of islands with which has a space complexity of **O(1)**.

**Time**

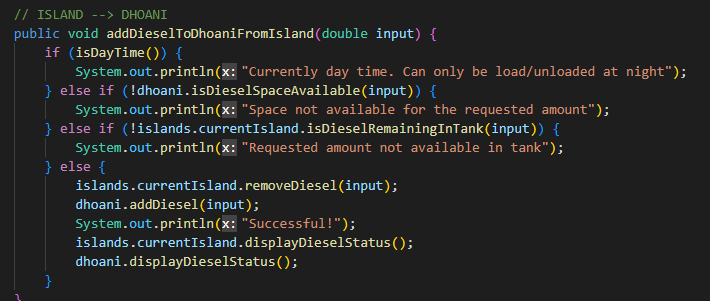
**Input**

**O(1)**

While a doubly linked list uses up more space, since a previous pointer needs to be added, it was necessary for dhoani to travel to two directions.

# System Architecture

* Main.java
  + responsible for displaying menu and calling appropriate functions
* Dhoani.java
  + Defines the storages and max capacities
  + Has the basic functions of only dhoani like add, remove, display, check space
* Island.java
  + Defines the storages and max capacities
  + Has the basic functions of only Island like add, remove, display, check space
* IslandLinkedList.java
  + Defines the structure of the doubly island linked list
  + Has travel functions as well as other functions related to traveling
* Controller.java
  + Neither the Island.java, IslandLinkedList.java, or the Dhoani.java interacts with each other, meaning they cannot interact with each other and swap items as these classes are not meant to perform function on their own.
  + Rather, a controller class is used to dictate the flow
  + Below is an example of a controller class function



Notice that the controller class call the individual functions from the Island.java, IslandLinkedList.java, and the Dhoani.java class. **dhoani.isDieselSpaceAvailable()**, etc..

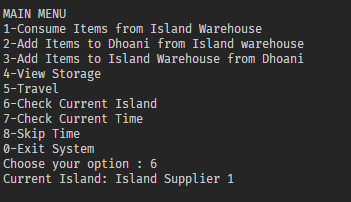
The user can interact with the system through a command line interface.

# Output and Testing

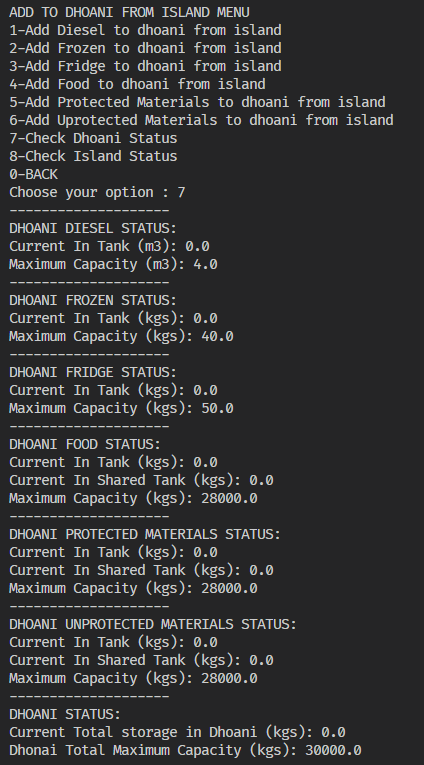
When the program is first run, the test data will also be loaded. The test data includes some of the islands already having some amount in storages. The dhoani will be initialized with empty tanks or storages and will start at the most west supplier island 1. The supplier island will also be initialized with large quantities in storages (to simulate infinity).

**Adding materials to dhoani from island.**

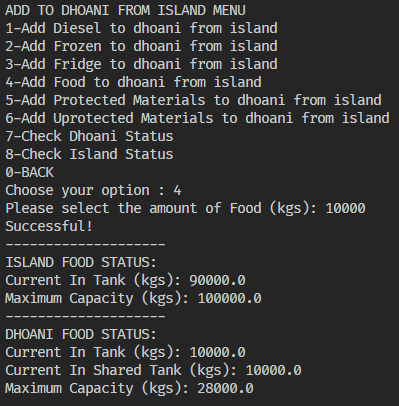
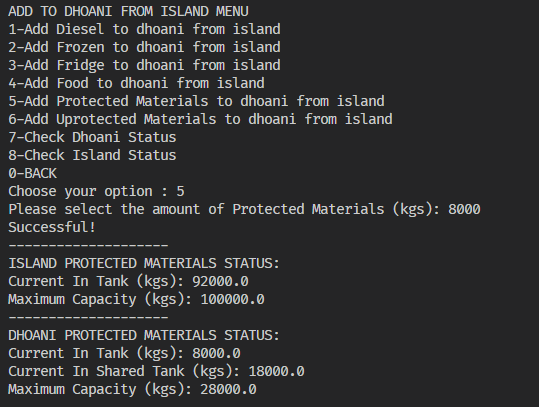
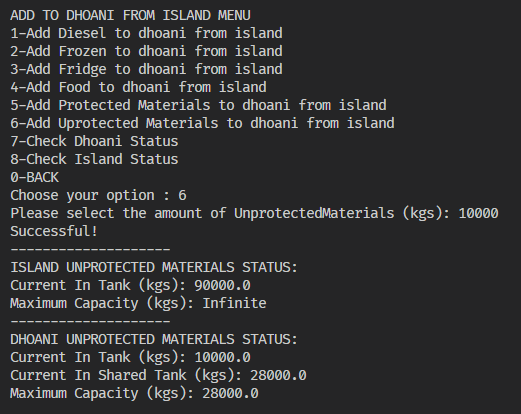
We start from supplier island.



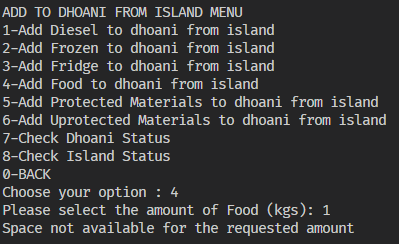
Currently the Dhoani is empty and will be loading items from supplier island one



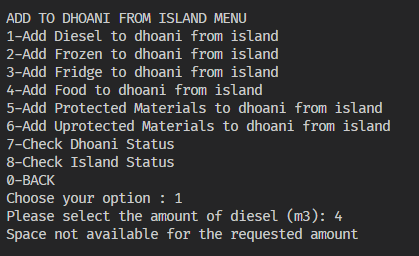
First, lets add the shared spaces in dhoani (Food, protected\_materials, unprotected\_materials) till the shared space for the 3 items becomes full.

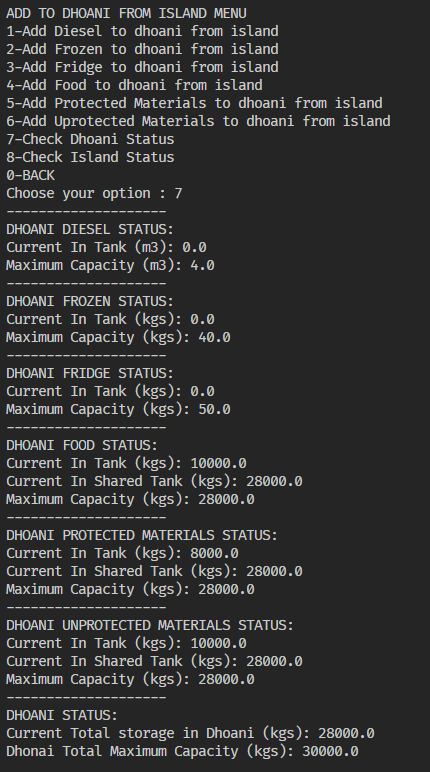
Above We see that indeed the 3 material share the same space in the dhoani.   
If we try to add beyond that we see that it is not allowed



Now let us add diesel to Dhoani from supplier island.

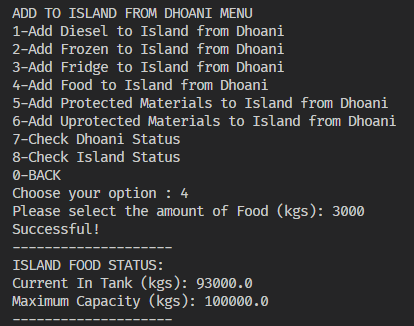
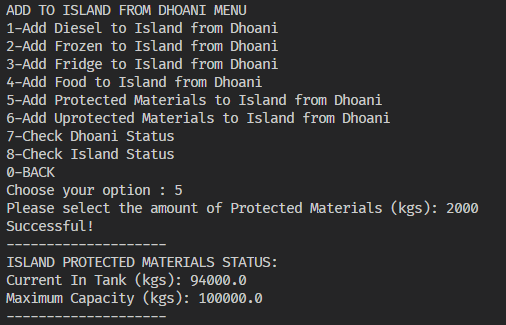
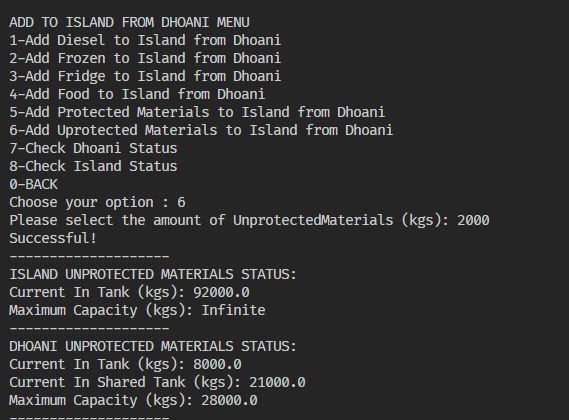


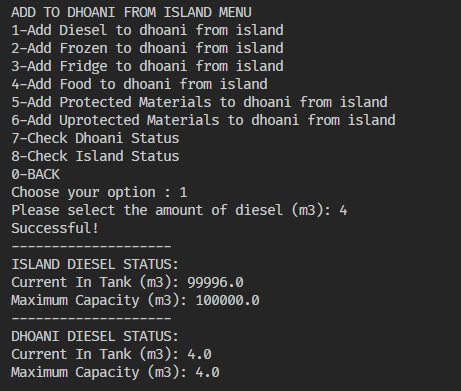
We see that there is no space available, although we did not load any diesel previously.  
To investigate, the user can check the dhoani status by choosing option 7 from the menu.



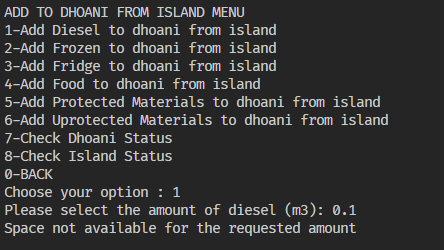
Upon investigating, it can be seen that the dhoani diesel space is available. However, the issue arises from the total dhoani space seen in the most bottom stats. The dhoani has a total capacity of 30,000 kgs. When the user tried to 4(m3) of diesel (which gets converted into kgs) its goes beyond the maximum capacity allowed for dhoani.

The user can transfer some items from dhoani to island to make space.



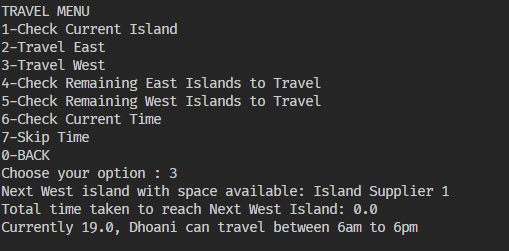
If the user tries to add further diesel, the system will prompt an alert message



Traveling to Next Island

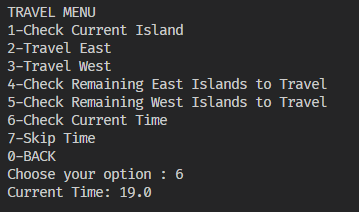
After getting supplies from the supplier island, the user can travel to next island from the travel menu

The user can try traveling west, but will be met by an alert message.

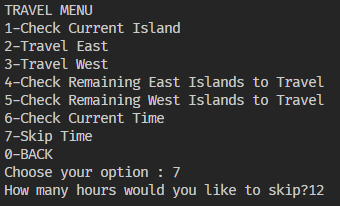
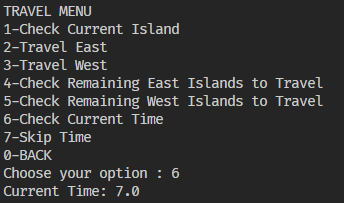


This is because it is currently night time, and the dhoani can travel between 6 am to 6 pm.

Current Time:

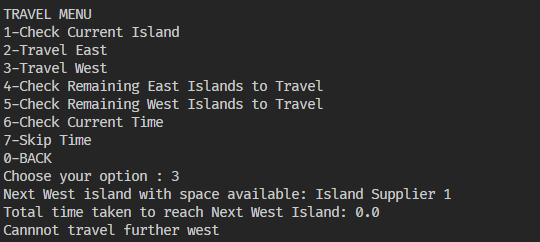


Time can be skipped by giving input in hours.

12 hours is skipped and current time is skipped to 7am

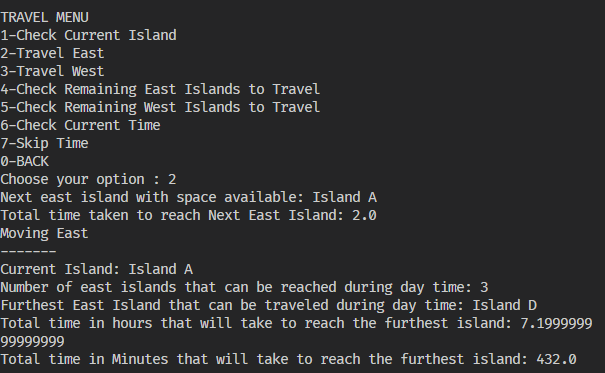
The user can try traveling west again, but will still be met with an error message.



This is because this is the edge island and the user has to travel east.

Note that the system will also display the next island that can be traveled as well as the time taken to travel to next island.

The user can travel to east successfully.



The system will display the next island (Island A) that can be traveled as well as the time taken to travel to next island before traveling.

After successfully travelling to the island, the system will also display the furthest island to the east that can be traveled during day time and the total number of time (in minutes and hours) to reach the furthest island since it is vital information that the captain of Dhoani requires.

DHOANI TO ISLAND

Now that we have reached island A, we can test removing items from dhoani and adding them to island warehouse functions.