Qualabs Challenge

Technical guide for the description of the methods and operations that solve the proposed challenges.

Common methods for the methods that solve the problems posed

1) LoadData

a. The method for loading files with the . json extension. This method is responsible for doing a search in the path it receives as an argument.

b. After having loaded all the data needed, we proceed to map all the json type structures against the user structure that is composed of other structures that help shape the complete object.

c. Once the previous process is completed, a user type array is filled and the name of the file of each loaded . json is added, this as an id to be used in later steps.

2) feedMaps

a. This method serves as a helper method to build the map type objects that will contain the relationship of each user with the module they use.

b. Using the features of maps in Go, which do not allow repeated keys, we can load unique keys and users associated with these keys (keys -> modules).

3) checkTotalModules

a. This method receives an array of type user, an array of action limited to the scope of the method is created in order to only use the length as a figure of merit with respect to the array of all modules in the system.

b. This method returns the size of the totalModule variable.

4) checkModuleInModules

a. This method verifies that before adding a new module, it is unique and does not appear in the array of total modules.

b. It returns true or false as appropriate. With this we know if it already exists and in that case it is not added.

Solution to challenge A

This method is in charge of solving challenge A, using secondary methods the variables necessary for the operation that will give the expected result are prepared.

It returns a variable of type byte array and an error. In the expected case, the array will contain the information ready to be formatted as a string and displayed on the console. In the case of an error, the error type variable will display the error associated with the step that provided it and the byte array will be null.

Solution to challenge B

This method is responsible for solving challenge B. Using the secondary methods, the variables necessary for the operation that will give the expected result are prepared.

This method implements a complex search algorithm in data structures. These types of algorithms are known as “backtracking” or exhaustive recurrent search algorithms.

Description of how backtracking algorithms work

Backtracking is exhaustive, which means that it tries to explore all possible solutions in a search space, discarding those that do not meet the restrictions as it progresses. It does not use additional knowledge or rules to reduce the search space beyond the explicit restrictions of the problem.

Restrictions are added to the search process that make possible the minimum combination of all users who use all modules.

Returns the minimum group of users that use all modules.