Assignment

GoLang Coding Exercise

Position

GoLang Software Engineer

Submission:

To submit the assignment, create a new repository named as - `[candidate name] - [assignment] - [position]`. You can use github, gitlab or bitbucket.

The repository should contain one `main.go` file, which executes two functions. Each function should point to the file containing the exercise solved.

Output:
When the main.go file is compiled and run, the output should clearly mention which exercise was it for, and the answer. Please use the `fmt.print` function to print your output.

Assignment 1

Binary Tree Assignment

A tree is a representation of a hierarchical structure. It's easy to imagine a tree by thinking about a family genealogy tree.

A binary tree is a tree where every node has max 2 children.

A binary search tree has the property of the left node having a value less than the value on the right node.

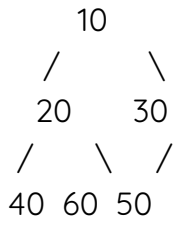
A binary search tree has the property of the left node having a value less than the value on the right node.

Write a package to generate a binary tree.

1. Write the function Insert() that takes the value as an input and adds nodes at the correct position into the binary tree.
2. Write the function InOrder() that takes the root node of the tree as input and returns a list containing the In-Order Traversal of the given Binary Tree.
3. Write the function PreOrder() that takes the root node of the tree as input and returns a list containing the Pre-Order Traversal of the given Binary Tree.
4. Write the function PostOrder() that takes the root node of the tree as input and returns a list containing the Post-Order Traversal of the given Binary Tree.

Expected Time Complexity: O(N).
Expected Auxiliary Space: O(N).

Example:



Inorder traversal: 40 20 60 10 50 30

Preorder traversal: 10 20 40 60 30 50

Postorder traversal: 40 60 50 20 30 10

Rules

1. We are really, really interested in your object-oriented or functional design skills, so please craft the most beautiful code you can.
2. We're also interested in understanding how you make assumptions when writing code. If a particular workflow or boundary condition is not defined in the problem statement, what you do is your choice.
3. You have to solve the problem using GoLang without using any external libraries to the core language except for a testing library for TDD.

Assignment 2

House Robber

You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed. All houses at this place are arranged in a circle. That means the first house is the neighbor of the last one. Meanwhile, adjacent houses have a security system connected, and it will automatically contact the police if two adjacent houses were broken into on the same night.

Given a list of non-negative integers nums representing the amount of money of each house, return the maximum amount of money you can rob tonight without alerting the police.

Example 1:

Input: nums = [2,3,2]

Output: 3

Explanation: You cannot rob house 1 (money = 2) and then rob house 3 (money = 2), because they are adjacent houses.

Example 2:

Input: nums = [1,2,3,1]

Output: 4

Explanation: Rob house 1 (money = 1) and then rob house 3 (money = 3). Total amount you can rob = 1 + 3 = 4.

Example 3:

Input: nums = [0] Output: 0

Rules

1. We are really, really interested in your object-oriented or functional design skills, so please craft the most beautiful code you can.
2. We're also interested in understanding how you make assumptions when writing code. If a particular workflow or boundary condition is not defined in the problem statement, what you do is your choice.
3. You have to solve the problem using GoLang without using any external libraries to the core language except for a testing library for TDD.