Longest Consecutive Sequence

Problem Statement

Given an unsorted array of integers nums, find the length of the longest consecutive elements sequence.

You must write an algorithm that runs in O(n) time complexity.

Example 1:

Input: nums = [100, 4, 200, 1, 3, 2]

Output: 4

Example 2:

Input: nums = [0, 3, 7, 2, 5, 8, 4, 6, 0, 1]

Output: 9

Constraints:

- 0 <= nums.length <= 10^5
- -10^9 <= nums[i] <= 10^9

Subarray Sum Equals K

Problem Statement

Given an array of integers nums and an integer k, you need to find the number of continuous subarrays that sum to k.

Example 1:

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

Output: 2

Example 2:

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

Output: 2

Constraints:

- 1 <= nums.length <= 2 * 10^4
- -1000 <= nums[i] <= 1000
- -10^7 <= k <= 10^7

GoKart Application

Objective:

The objective of this task is to help you understand how to use JDBC to work with a database efficiently. You will learn how to perform complex queries involving joins, grouping, and aggregation, as well as updating data in the database.

Instructions:

Setup:

- Ensure you have a database server set up and JDBC drivers installed.
- Create a database and tables using the schema provided below:

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(100),
    ContactName VARCHAR(100),
    Country VARCHAR(50)
);

CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10, 2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    UnitPrice DECIMAL(10, 2)
);

CREATE TABLE OrderDetails (
    OrderID INT,
    ProductID INT,
    Quantity INT,
    PRIMARY KEY (OrderID, ProductID),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

- Populate the tables with sample data.

Tasks:

1. Write a Java program using JDBC to fetch and display a list of all customers who have placed an order in the database, including their CustomerName, ContactName, and Country.
   (Hint: You'll need to perform a join between the Customers and Orders tables.)
2. Write a Java program using JDBC to calculate the total sales amount for each country and display the results. The results should include the country name and the total sales amount.
   (Hint: You'll need to perform a join between Orders and Customers and use a GROUP BY clause.)
3. Write a Java program using JDBC to update the unit price of a specific product in the Products table. Allow the user to specify the product name and the new unit price.
4. Write a Java program using JDBC to delete a specific order by its OrderID. Allow the user to specify the OrderID.