iOS Interview Questions:

- Deletion rules in Core Data
    - **Nullify:** In Managed Objects relationship is make nil so the object keeps there only relationship is nullify
    - **Cascade:** All objects that in relationship with the deleted object will deleted(if category is deleted all items in category will delete)
    - **Deny:** If a object is in relationship with other objects then it deny to delete that first need to delete all those object that are in relationship
    - **No Action:** Object is deleted, all related objects that are in relationship are there. Generally not preferred.
- Core Data is thread safe?
    - No
- How to make core data thread safe?
    - To make thread safe for ui work use on ‘MainQueueConcurrencyType’ and ‘PrivateQueueConcurrencyType’ for background work
    - Passing data between context use Object ID of the managed object as this is the thread safe
- Core Data Migration
    - LightWeight
        - Renaming attributes, adding new attributes creating new entities will comes under this, it is defined under ‘NSPersistentStoreCoordinator’
    - HeavyWeight
        - Splitting an entity or merging or changing the relationship between entities comes under this.
        - Done by Core Data Model Editor in Xcode
- Safety in coredata
    - Ensuring data integrity, managing concurrency properly and handling errors effectly.
- What are Core data faults?
    - They act as lightweight placeholders that delay data loading until absolutely necessary
- Protocol type
    - Optional
        - Can be achieve by the making extension of that protocol and give default implementation
    - Required
        - Have to implement all methods inside the class
- Array vs Set
    - Arrays
        - Arrays are ordered collections of the same data type. They contain duplicate values.
    - Set

        - Sets are unordered collections of the same data type and they do not contain any duplicate values.
- Unit Test
    - XCTest
    - @Testable
    - SUT(System under Test) Objects
    - XCAssertNil, XCAssertTrue, XCAssertFalse etc
    - How to write test cases for Async Task
        - Wait for expectation to be fulfill with timeout
- Enum
    - Can give you capability to write the some finite number of cases
- Dependency Injection
    - It is a design principle in swift to achieve the modularity and testability of a system.
- Clouser
    - Can give you the ability to pass a function as a parameter and it is a promise to the system it will execute before the function.
        - NonEscaping -> Used in synchronous task
        - Escaping -> Used for asynchronous
    - Inline/Trailing Closure:
        - **Inline Closure** are those which can be defined directly in function calls. Right where you need it.
            - let sortedNumbers = numbers.sorted(by: { (a: Int, b: Int) -> Bool in return a > b })
        - **Trailing Closure** is written outside of the parameter of the function generally used when closure is the last argument in the function.
            - let sortedNumbers = numbers.sorted { (a: Int, b: Int) -> Bool in return a > b }
- How to use swift code inside Objective c
    - By using bridging headers
    - And then in top you need to import that swift file and use it in objective c code
    - And to call the swift functions and classes in objective c class need to call it by @Objc as by default Swift not expose his functions to the objective c
- Application Binary Interface.(ABI)
    - It's a some set of rules that define how the swift code is compile to the binary code
- Configuration of URLSession
    - Like timeout, cache policy and many more things
- Type of configurations in URLSession
    - Default
        - Use this for most networking tasks that do not require background execution or special privacy handling.
    - Ephemeral

        - Use this when dealing with sensitive information or when you don't want to store any session data on disk.
    - Background
        - Use this for tasks that need to continue running when the app is in the background, like file uploads or downloads.
    -
- Race Condition
    - Read or write operation performed on a shared resource by multiple points at same time then race condition occurs
        - .barrier will solve this issue
        - DispatchSemaphore will solve this issue
- DispatchSemaphore
    - wait()
    - signal()
- How does APNS works internally
    - App -> Mobile -> APNS
    - APNS -> Mobile(token) -> App -> Server -> send notification to app
- Authentication token type
    - Bearer token
    - JWT (JSON Web Tokens)
    - OAuth Token
    - MAC(Message Authentication Token)
    - Hawak Token
    - API Keys
    - Session Tokens

- Carthage, SPM and CocoaPods
- Unowned Vs weak
    - To help in break the retain cycles
- Protocol is is value type or reference type
    - They are not value of reference type they are just blueprint for methods, variables and other requirement that suited for a task
- How to make singleton thread safe?
    - While using shared resource apply lock and once task done apply unlock this way we make it thread safe
    - By using GCD DispatchQueue custom one

- Type of Error Handling
    - Synchronous
    - Asynchronous
    - Throwing functions
    - Optionals
- SOLID Principles
    - **Single Responsibility:** Each task have separate class

    - **Open close:** Open for extensions and close for modifications
    - **Liskov substitution:** Child class implement in such a way do not violate the implementation of parent class
    - **Interface Segregation:** Any client should not be forced to implement all methods that are irreverent to them.
    - **Dependency inversion:** High level components should not directly depend on low level implementation.
- Multithreading
    - Achieve by GCD
    - Operation Queues
- GCD
    - Main Queue
        - Execute serial manner
    - Global Queue
        - Quality of service(QOS)
            - **User Interactive:** Perform a task that requires immediate completion
            - **User Initiated:** Perform a task that the user is waiting for
            - **Utility:** Perform a long-running background task
            - **Background:** Perform a non-critical background task
        - Custom Queues
            - By Default a custom queue is a serial one
- Operation Queues
    - Block Operations
    - Add Dependency
    - Define number of concurrent task need to execute once
    - Made top on of GCD
- Serial queue vs Concurrent queues
    - Serial queues execute one after completion of existing
    - Concurrent quest runs the task parallel by using of context switching and time slicing
- Dispatch Group
    - .enter
    - .leave
    - .notify(): Recommended way
    - .wait()
- Dispatch Work Item
    - .cancel()
    - Store in global variable
    - Add a delay
- High Order functions
    - Filter
    - Map, Compact Map
    - Reduce

- Pyramid of dom
    - Async await will solve this
- Parse Json:
    - JSONDecoder(SnakeKeyDecodingstreategy, dateDecodingStreategy)
    - JSONSerializer
- KVC/KVO
    - KVC
        - @Dynamic @objc will make it capable the KVC any class object type
        - Due to this get set will access
    - KVO
        - Will observe the any change in a variable it is observing for a change
- Optionals: Optional is enum type
    - Some
    - None(nil)
- Optional Chaining
    - Done by ? after each variable at last use cascade nil
- Unwrap Optionals
    - Guard let
    - If let
    - Force unwrap
    - Check != nil
- Copy on write
    - Array in swift have implicit copy on write feature
    - Means if a object is created from and existing object it's address is going to be same until any changes do not perform on the new object that is created
- Generic
    - That can work with any data type without sacrifice of it;s type safety
- Difference in generic and any
    -
- Struct vs Class
    - Struct value type Class are reference type
    - Struct do not have deinit, Class have deinit
    - Struct does not have inheritance, Class have inheritance(Multiple inheritance not supported)
    - Struct stored in Stack, Class stored in heap
    - Struct faster than Class
- When should you class over struct
    - If we need to communicate with the objective c classes we need to use class over struct as struct are not in objective c
    - If we need to apply inheritance then use classes as struct do not support inheritance
- Run Loops
- Application life cycles
    - Not running

    - Inactive
    - Active
    - Background
    - Suspended
    - Foreground
- View Life cycles
    - LoadView
    - View Did Load
    - View Will Appear
    - View Will Disappear
    - View Did Appear
    - View Did Disappear
    - viewWillLayoutSubviews
    - viewDidLayoutSubviews
- SwiftUI Life Cycles
    - OnAppear
    - One Disappear
    - task
- Lazy keyword
    - Stored property
- Defer
    - Calls before return of function
    - If Multiple differ statements are there it execute from bottom to up manner
- Method Dispatch
    - Static Dispatch: Value type
    - Dynamic Dispatch: Reference type
        - Have a witness table that stores the index of functions of classes
    - Message Dispatch(Method Swizzling)
        - Does not have witness table
- Protocol(Blueprint of classes)
    - Protocol Extensions
    - Protocol method type
    - Protocol Composition
    - Associated protocol
- Any vs AnyObject
- Delegates
    - Data passing mechanism in backward directions
    - One to one way
- Initializers
    - Designated
    - Convenience
    - Failable
    - Required
- Async/Await

    - Perform asynchronous task in more readable way by throwing errors efficiently
- Actor(MainActor)
    - Similar to classes
    - Does not support inheritance
    - Thread safe
- Combine
    - SwiftUI apple framework for reactive programming
    - Uses Publisher and subscriber model
- RxSwift
    - Reactive programming
    - Third party framework
- Recursion
    - Inside a function call itself until condition won't break
- Git rebase and merge
    - `git merge`: Combines branches with a new merge commit, preserving the history of both branches.
    - `git rebase`: Re-applies commits from one branch onto another, creating a linear history without merge commits.
- Git Squash
    - Git squash is a technique used to combine multiple commits into a single commit.
- Cherry Pick
    - Selecting specific commits from one branch and applying them to another branch.
- Atomic and nonAtomic in Objective C
    - Atomic
        - Thread safe
    - NonAtomic
        - Not thread safe
- We have two view arranged horizontally in vertical mode how to placed vertically in landscape mode from storyboard only
    - Size Classes wAhC will do that
- Apply constraints from code
- Leading constraint from code
- Share extension
    - How to add our app in the share extensions options once any app wants to share images from them.
        - Add a share extension for your app, this is a separate target within your app to handle incoming share data.
        - Configure the share extension -> Define the maximum number of image support by our app
        - Handle the Received Data in Your Share View Controller -> Get the url or image need to handle it accordingly
- IOT(Internet of Things)
    - Works on MQTT(Message Queue Telemetry Transport) protocol

    - Lightweight and ideal for the works on low bandwidth and low energy
    - Components
        - Publisher(Raspberry pie)
        - Subscribers(ESP32)
        - Broker(Mosquito)
    - Works on Pub-Sub Model
    - Subscriber needs to subscribe to a topic that is published by the publisher.
    - Topic
        - “/home/bedroom/light” ->
        - “/home/+/light” ->
        - “/home/*” -> Wild card
    - Ios library used for this CocoaMQTT
        - Methods
            - Connect
            - Disconnect
            - didReceive Messages
            - Subscribe

Programing questions:

- Write a function to find the index of first non repeating character from given string “rajkumar”
- Demonstrate the data passing via closure between two controllers.
- Find non repeating items from an array of integer
- Factorial by recursion
- Exponential by recursion
- Find string is valid or not “(Show)h(of)” if braces start and end correctly it is valid if not then invalid
- var array : [Any] = [1,2,3,4,true,false,"Shashi", "interview"]
    - Write an high order function that written only string value
        - array.compactMap({$0 as? String})
- Write and example for serial queue and a concurrent queue
    - Serial Queue
        - let serialQueue = DispatchQueue(label: "com.serai.Queue") // By default a custom queue is a serial queue until it defined as a concurrent
    - Concurrent Queue
        - let concurrentQueue = DispatchQueue(label: "com.serai.Queue", attributes: .concurrent)

Algorithms:

1. Kruskal's Algorithm
2. Run Loop Encoding algorithm