Interview Questions

# First Interview Questions

1. **LRU Cache**
   1. We want to design an LRU cache with a known capacity and two operations:
      1. Set(Key, Value) -> Null
      2. Get(Key) -> Value
   2. The cache eviction policy is LRU == least recently used == we kick out the key that was accessed (either Get or Set!) the longest time ago. The **number** of usages is irrelevant.
   3. The final solution can be O(1) on average, by using a hashmap/dictionary and a doubly-linked-list to track the eviction order
   4. Once they use a hashmap/dictionary/javascript object, the candidate should be asked to explain the complexity - O(1) on average, how it works under the hood and when it’s not O(1) in real life
   5. They should be asked to explain the flow from start to finish (eventually) for get and set operations, including what they store in each data structure, and how the linked list is manipulated (and so why it has to be doubly-linked)
   6. Note whether they picked up subtle points, like updating the value of an existing key.
2. **Rest API/DB**
   1. We are making a new product design - computerizing a small shop.  
      We’ll be discussing various customer needs and eventually suggesting a client-server application for a cash register.
   2. The outline is as follows:
      1. Introduce the story
      2. Discuss various areas of running a shop where software can help
      3. Focus on building a cash register - specifically the user-story of a purchase of physical products
         1. Collect requirements
         2. Discuss what will be the responsibility of the front and back end
         3. Design REST APIs (method+url+params+body+return type)
         4. Select a database and design database schema (columns, types, PKs, FKs, Indices)
         5. Go over the scenario of a purchase, producing an invoice
         6. Go over the scenario of retrieving an invoice (what if the price got updated in the meantime?)
         7. Stretch question if there’s time - price history query (how can we find the list of products that were updated in the last 3 months)
3. **Search Engine**

**Definition**

We have a lot of text files that are immutable. The user input is a word and we will list all the file names that contain the given word (full word only).

How can we implement it in the most efficient way?

Once we have it, how can we add a feature that the input is 2 words and we list all the file names that contain these 2 words.

**Solution**

First we will scan all files and use hash to store each word with a list of all the file names we found. Given a word we can list the file names on O(1).

For the second part we will need to get the 2 lists of file names and find the names in both lists. Naive approach will be O(N\*M) (given N and M are the length of each list). We can improve that by saving these lists sorted and then we can find the common names in O(M+N) by running in parallel on both lists.

1. **Smart Dictionary**

**Definition**

Design a smart dictionary with the following API

1. *set(key,value)* - set a key with a value
2. *get(key)* - return the value of the key
3. *setAll(value)* - define that *get(key)* will return the value to all key unless we set it after the setAll

**Solution**

*set(key, value)* - save the value and the creation timestamp in a Hash

*setAll(value)* - save the value in a different variable and the creation timestamp.

*get(key)* - if key does not exist, check if we have a setAll variable and return it.

If key exist, check if the timestamp in the Hash is after the timestamp in the setAll variable and return the most updated value

1. **General Knowledge Questions**
   1. What is https? What is it used for? How does it work?
   2. What are cookies? How can we use them?

**6. Garbage Collection**  
 Suggested question about garbage collection - [link](https://docs.google.com/document/d/1oEWkgASUg4zh8621LRSxP1Zide0Vqx0orpgfdLLEOI4/edit?usp=sharing) (draft, needs review)

# 

# Second Interview Questions

1. **spOT   
   TBD**(helper drawing here - [draft](https://excalidraw.com/#json=HHvOn0wnsPdRcjaHAIZdp,XOlYmvhHMu97RoOYhoVVOw)!)
2. **Event Manager**

**Definition**

Define a system that will have the following API

1. *register(eventName, callback function)* - register an event with a function that will be invoked when fire the event
2. *fire(eventName)* - invoke the function
3. *unregister(eventName)* - unregister the event

**Solution**

Few solutions for this, it is a language dependent but need to pay attention to the following issues:

1. Does fire running the function in a sync mode or not
2. What happens if we register the same event name on multiple functions? How do we handle delete in this case?
3. How can we handle errors on the invoked function?

Need to see the technology the candidate choose to use: Messages/Queue, Redis, DB etc

# Suggestions for review:

1. <https://www.geeksforgeeks.org/largest-sum-contiguous-subarray/>