Assignment 3 : Basic Data Structure

1. Implement a **Self Balancing Binary Search Tree** with following supported operation : **(3 Marks)**
    a. find(key) : returns true if key is present else false
    b. insert(key) : insert a new key
    c. remove(key) : remove an existing key
    d. order_of_key(key) : returns the order of the key compared to the existing elements i.e., how many elements are smaller than key
    e. get_by_order(k) : returns the k'th element among the existing keys
2. Implement a **Hashmap** with following supported operation : **(3 Marks)**
    a. find(key) : returns true if key is present else false
    b. insert(key, value) : insert a new {key, value} pair
    c. remove(key) : remove an existing key
    d. Implement both open addressing (Linear and/or Quadratic probing) and separate chaining techniques for collision handling
3. Implement **LRU Cache** with following functionalities : **(2 Marks)**
    a. LRUCache(capacity) : Initialize the LRU cache with positive size capacity.
    b. get(key) : Return the value of the key if the key exists, otherwise return -1.
    c. put(key, value) : Update the value of the key if the key exists. Otherwise, add the key-value pair to the cache. If the number of keys exceeds the capacity from this operation, evict the least recently used key.