<u>Lesson-11 Hash table – Class Notes</u>

A hashtable (or hash table) is a data structure that implements an associative array, a structure that can map keys to values. It uses a hash function to compute an index into an array of buckets or slots, from which the desired value can be found.

Array we follow index based approach to store and retrieve. Index is int value

Hash Table concept

Index based approach - index identified using some hashcode

To store on data in to the table you need to know two things

1. Key
2. Find index to store hashcode of the key % table size

Key is an object type

Key is an integer you can directly apply key % Tablesize
Key is an object you can apply Math.abs(key.hashCode()) % tablesize

Advantages

- Fast Insertion and deletion, retrieval

Disadvantages & Key Points

- You cannot do sorting
- Always you need a table capacity
- Prefer table capacity as prime number
- Declare the table capacity with more than (25%) of the expected requirements
- Deal any one of the Collision Techniques (Linear Probing, Quadratic, Double hashing or Separate chaining)

Collision:

- A collision occurs when two different keys are hashed to the same index or location in the underlying array.
- In other words, the hash function generates the same hash value for two distinct keys, causing both keys to be mapped to the same slot in the array.

Common strategies to handle collisions include:

1. **Chaining**: Each slot in the array points to a linked list of entries that hash to the same index. When a collision occurs, the new entry is simply added to the linked list at that slot. In Java's API HashMap, if the size of one of these linked lists gets to be too large (default = 8), the linked list is converted to a red-black tree for the efficient operations.

https://yongdanielliang.github.io/animation/web/SeparateChaining.html

2. **Open Addressing**: When a collision occurs, the hash table looks for another empty slot within the array to store the new entry. This is done through various probing techniques such as linear probing, quadratic probing, or double hashing.

https://yongdanielliang.github.io/animation/web/LinearProbing.html
https://yongdanielliang.github.io/animation/web/QuadraticProbing.html
https://liveexample.pearsoncmg.com/dsanimation/DoubleHashingeBook.html

Hash based Java APIs

1. HashSet - Set Interface
2. Hashmap - Map Interface
3. Hashtable - Map Interface

First Rule About Creating Keys for a Hashtable: *To use an object as a key in hashtable, you must override* equals() *and* hashCode().

Creating Unmodifiable Collection

https://docs.oracle.com/en/java/javase/17/core/creating-immutable-lists-sets-and-maps.html#GUID-1222F8A3-7EC0-4E49-9B75-C3B263F9A1BB

JDK 8: Uses Collections.unmodifiableXXX methods. Requires creating a modifiable collection first and then wrapping it in an unmodifiable wrapper.

JDK 9 and Later: Uses new factory methods like List.of, Set.of, and Map.of. These methods directly create unmodifiable collections, simplifying the creation process and reducing boilerplate code.

Advantages: Thread Safety, Immutability, and Security.

Refer: lesson11.unmodifiablecollection