# Questions:

* Hash table vs Hash Map?
* Hash-code vs equals?
* Rehashing?

# Notes:

* Hash-code is calculated by hash-function using the “key”. i.e.
* Equal keys should have equal hash-codes.
* Hash function should be efficient to compute.
* If two keys map to the same location – a collision is said to have occurred. A good hash function will spread two keys into different “slots”. If collision occurs then two keys are in the same slot, but in the form of a list
* Should avoid using **mutable objects** as keys, because if the key mutates, then it’s position will change, and couldn’t be look up.
* Objects are stored in array locations based on hash code of key.

# How to design a hash function (for strings):

* A hash function should examine all **characters** in the string
* A hash function should give **large range of values** (so that there is little **collision**), and not let one character dominate.
  + For instance, if we **cast characters to integers and multiply them** and one of them ended up being “0”, the entire product will become 0, therefore it is not the best way to compute hash code.
* A rolling hash function is important.
  + IF, one **character** is deleted from the **front** and added to the **end.**
    - THEN, the new hash-code can be computed in O (1) time.

# Uses of hash tables:

* A good way to represent a “set of strings”
* A trie is better when the set of strings is “dynamic” in nature
  + Nodes in the tree do not represent the “key”
  + Node’s position in the tree, defines the key which it is associated with

# Examples of usage of hashtables

* An application that benefits from “algorithmic advantages” of a hash table, i.e. Anagrams
  + Get a set of words – like a list
  + Returns group of anagrams – like a list of groups of anagrams (which is also a list)
* Design of a class that can be used in a hash table.

# Hash table libraries

* Hash tables have two types
  + Hash map
  + Hash set
* Hash tables have best “lookup, “insert” and “delete” time complexity
  + O (1) in average case
  + O (n) in case the hash table has to be resized.