**Chapter 01- Arrays and Strings**

1. **Hash Tables:** a data structure that maps keys to values for highly efficient lookup. There are a number of ways to implement this. This is a simple but common implementation where we use an array of linked lists and a hash code function. To insert a key (Which might be a string or essentially any other data type) we do the following:
   1. First compute the key’s has code, which will usually be an int or long. A long is a datatype used in certain programming languages to store a single 64-bit signed integer. Two keys could have the same hash code, as there may be an infinite number of keys and a finite number of ints.
   2. Then, map the hash code to an index in the array. This could be done with something like hash(key) % array\_length. Two different hash codes could, of course map the dame index.
   3. At this index there is a linked list of keys and values. Store the key and value in this index. We must use a linked list because of collisions: You could have two different keys with the same hash code, or two different hash codes that map to the same index.
   4. To retrieve the value pair by its key, you repeat this process. Compute the hash code from the key, and then compute the index from the hash code. Then search through the linked list for the value with this key.
2. **ArrayLists:** are automatically resizing arrays. When the array gets full, it doubles in size and the time it takes to implement the growth of the array is amortized over all the other calculations. This is an essential data structure for interviews.
3. **StringBuilder:**