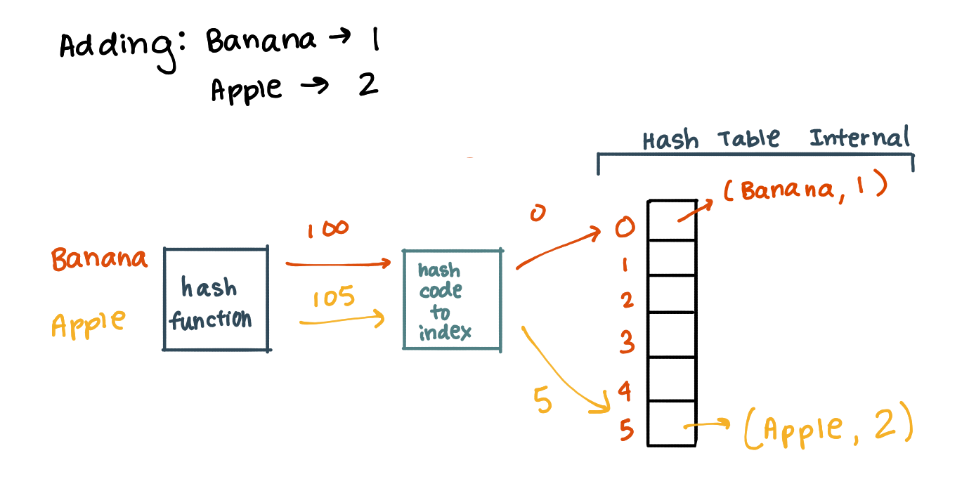
*Hash Tables*

A hash table is an abstract *dictionary* that stores ***values*** and each of these values has a ***key***. Hashing is a technique or process of mapping keys, values into the hash table by using a hash function. It is done for faster access to elements. The efficiency of mapping depends on the efficiency of the *hash function* used.

Each key is ***unique*** and we can transform keys into integers to use them as an index. A ***collision*** occurs when two or more keys are assigned to the same index.



*Huffman Coding*

Huffman coding is a ***lossless*** ***data*** ***compression*** algorithm that uses the *Greedy* *choice* *algorithm*. The idea is to assign variable-length *codes* to input characters and lengths of the assigned codes are based on the *frequencies* of corresponding characters. The most frequent character gets the smallest code and the least frequent character gets the largest code. This works by creating a *Huffman* *Tree* that needs to be store alongside the compressed data.

*Huffman Tree Algorithm:*

1. Create a node for each character and label each with the frequency
2. Arrange these nodes in ascending frequency
3. Take the first two nodes and join them then the parent node should be labeled with the sum of these two nodes
4. Insert the created node into the ordered list and then repeat from step 3

*Properties:*

- the *leafs* of the tree will be the *characters*

- the *root* should be the total *number* *of* *characters* from the input

- the *left* arrow is labeled 0 and *right* is 1

- to read the ***code*** start from the top and trace to the character reading off 1s and 0s

- the more *skewed* the tree, the *better* the compression of the data because it means that a lot of the data is being repeated

