coursera

Lab Help

Hash functions and multiplanes

In this lab, we are going to practice the most important concepts related to the hash functions explained in the videos. You will be using these in this week's assignment.

A key point for the lookup using hash functions is the calculation of the hash key or bucket id that we assign for a given entry. In this notebook, we will cover:

- Basic hash tables
- Multiplanes
- Random planes

Basic Hash tables

Hash tables are data structures that allow indexing data to make lookup tasks more efficient. In this part, you will see the implementation of the simplest hash function.

In [1]:
```
import numpy as np                # library for array and matrix manipulation
import pprint                     # utilities for console printing
from utils_nb import plot_vectors # helper function to plot vectors
import matplotlib.pyplot as plt   # visualization library

pp = pprint.PrettyPrinter(indent=4) # Instantiate a pretty printer
```

In the next cell, we will define a straightforward hash function for integer numbers. The function will receive a list of integer numbers and the desired amount of buckets. The function will produce a hash table stored as a dictionary, where keys contain the hash keys, and the values will provide the hashed elements of the input list.

The hash function is just the remainder of the integer division between each element and the desired number of buckets.

In [2]:
```
def basic_hash_table(value_l, n_buckets):

    def hash_function(value, n_buckets):
        return int(value) % n_buckets

    hash_table = {i:[] for i in range(n_buckets)} # Initialize all the buckets i

    for value in value_l:
        hash_value = hash_function(value,n_buckets) # Get the hash key for the g
        hash_table[hash_value].append(value) # Add the element to the correspond
```