

What I had so far

February 1, 2021

```
[ ]: class Node:
    def __init__(self, data, next_node):
        self.data = data
        self.next_node = next_node

class HashTable:
    def __init__(self, n, hash_func):
        self.mylist = [None] * n
        self.hash_func = hash_func

    def get(self, key):
        location = key % len(self.mylist)
        nodes = [self.mylist[location]]
        #sublist = [nodes]
        temp = []
        counter = 0
        if nodes == None:
            return None
        else:
            for x in range(len(nodes)):
                temp.append(x)

                #temp.append(sublist[x].next_node)
            return temp

            # temp = []
            # for x in range(len()):
            #     temp.append(str(x.next_node))
            # return temp

            return nodes.next_node
    def insert(self, key, value):
        location = key % len(self.mylist)
        nodes = [self.mylist[location]]
        #sublist = [nodes]
```

```

    if nodes == None: #empty
        #nodes.append(Node(key, value))
        self.mylist[location] = Node(key, value)
    else:
        #sublist = [nodes]
        for x in range(len(nodes)):
            if key == x:
                nodes.insert(x, Node(key, value))
                #sublist[x].next_node = value
            else:
                nodes.append(Node(key, value))
        self.mylist[location] = nodes
        #return str(x.next_node)

def remove(self, key):
    location = key % len(self.mylist)
    nodes = self.mylist[location]
    if nodes == None:
        return None
    else:
        value = deletevalue = nodes.data
        if value == key:
            self.mylist.pop(deletevalue)
            return value

# simple hash function for testing
hash_mod = lambda key, n: key % n

#table size
n = 3#10

table = HashTable(n, hash_mod)
print(table.insert(0, 'a')) # slot 0
print(table.insert(3, 'b')) # slot 0 - collision
#print(table.insert(2, 'c')) # slot 2
print(table.insert(0, 'aa')) # overwrite previous value for key 0
#print(table.insert(6, 'd')) # slot 0 - collision
print(table.get(0)) # should print 'a'
#print(table.get(1))
#print(table.get(2))
#print(table.remove(0)) # should print 'a' and remove Node with key 0

```