Datastructures hashing with chaining

April 20, 2021

[17]:

```
hash_table = list([0 for i in range(7)])

def get_key(data):
    return hash(data)

def hash_func(key):
    return key % 7

def save_data(data, value):
    index_key = get_key(data)
    hash_address = hash_func(index_key)
    if hash_table[hash_address] != 0: #chaning
        for i in range(len(hash_table[hash_address])):
            if hash_table[hash_address][i][0] == index_key:
                hash_table[hash_address][i][1] == value
                return
        hash_table[hash_adress].append([index_key, value])
    else :
        hash_table[hash_address] = [[index_key, value]]

def read_data(data):
    index_key = get_key(data)
    hash_address = hash_func(index_key)

    if hash_table[hash_address] != 0:
        for i in range(len(hash_table)):
            if hash_table[hash_address][i][0] == index_key:
                return hash_table[hash_address][i][1]
    else:
        None
```

[23]:

```
save_data('qwer','1234')
save_data('fjdo','12889596')
save_data('qweji','288483')
```

```
a = get_key('qwer')
print(get_key('qwer'))
print(hash_func(a))
hash_table
```

```
7390549243261874227
6
```

```
[23]: [[[6740550256990479776, '12889596']],
        0,
        0,
        [[4992412736238370621, '288483']],
        0,
        0,
        [[7390549243261874227, '1234']]]
```

```
[ ]:
```