Hashmaps Worksheet

# Review Questions

## How hashmaps work.

1. What does a hashmap do? What are the methods that the client will use?

Hash Map are data structures that store key - value pairs. Given a key, we could look up the value in O(1) runtime.

The most often used methods are get, put, remove and contains.

1. Describe how a hashmap works concisely. Use less than 150 characters.

Suppose we have a hashmap with size M. If we want to look up a key whether in the hashmap or not, first we should find the hashnumber of the key with hash() function. Then we should find the bucket number, the simple way to find it is to mod hashnumber with the size of hashmap( hash(key) % M). Once we find the bucker number, we could check if the key is in the bucker or not.

1. What is a hash function?

A hash function is a function that maps a key to a integer number.

1. What is the most important quality of a good hash function?

A good hash function should meet three requirements: Deterministic, Efficient, Uniform.

1. Please write a good hash function for the following keys. Keep in mind the property you listed in (4)
   1. Strings

def hash\_string(string\_key):

N = 50

hash\_value = 0

for char in string\_key:

hash\_value += int(char)

return hash\_value % N

* 1. Float numbers

def hash\_float(num):  
 N = 50

float\_str = str(num)

for char in float\_str:

hash\_value += int(char)

return hash\_value%N

* 1. The following cat object.

class Cat:

def \_\_init\_\_(self, name, age):

self.name = name # string

self.age = age # int

def has\_cat(self):

N = 50

hash\_value = 0

for char in self.name:

hash\_value += (int(char))

hash\_value += self.age

return hash\_value % N

1. What is the runtime of a hashmap’s get method in the average case? O(1)
2. What is the runtime of a hashmap’s insert method in the average case? O(1)
3. What is the runtime of the insert function in the worst-case?O(n)
4. What type of input would give you the worst-case runtime for a hashmap? Describe this one in English.

All the inputs have the same hashnumber.

1. Please provide an example of the worst-case input (let’s say for size N = 10) for each of the hash functions on one of the hash functions you provided in (5)

5(a). If the input strings have the same combination, this will lead to the same hash number like abc, cab, bac, cab.

5(b). If the sum of all the digits of a float number is same with another float number, it will end with hash collision like 3.14, 4.13, 43.1

5(c). If the sum of the name and the age of a cat is same with another one, it will end with hash collision like cat1(“B”, 5 ) , cat2 (“B” , 5) or cat1(“B” , 5), cat2(“C” , 4), cat 3(“A”, 6)

1. What are the tradeoffs between linear probing and separate chaining? What types of inputs will have better performance for one over the other? Please provide a sample input for each example.

I am not sure how to answer this question, but I don’t think linear probing is a good idea.

# Leetcode Problems for This Section

* Spend only 45 minutes trying to devise a solution to each problem (everything except coding)
* Please [reference our guide on how to approach practice exercises in Teachable.](https://codebreakers1.teachable.com/courses/codebreakers-training-vault/lectures/14591190)

Complete the following Leetcode Problems and add them to your LC Review Schedule.

|  |
| --- |
| 1. [Design Hashmap](https://leetcode.com/problems/design-hashmap/) |
| 1. [Leetcode’s LRU Cache](https://leetcode.com/problems/lru-cache/) |