CS 1.2: Intro to Data Structures & Algorithms

Hash Table Worksheet

Name: Ayush Jain

Q1: What are the **3 ingredients** necessary to build a **hash table** data structure (with chaining)?

1. A Hash Function that calculates a fixed number for each input key.
2. An array to store several buckets, each with a unique index in range `[0 ... b-1]`.
3. Several linked list structures so we can store multiple entries in each bucket.

Q2: What are the steps required to **add a new entry (key-value pair)** to a hash table?

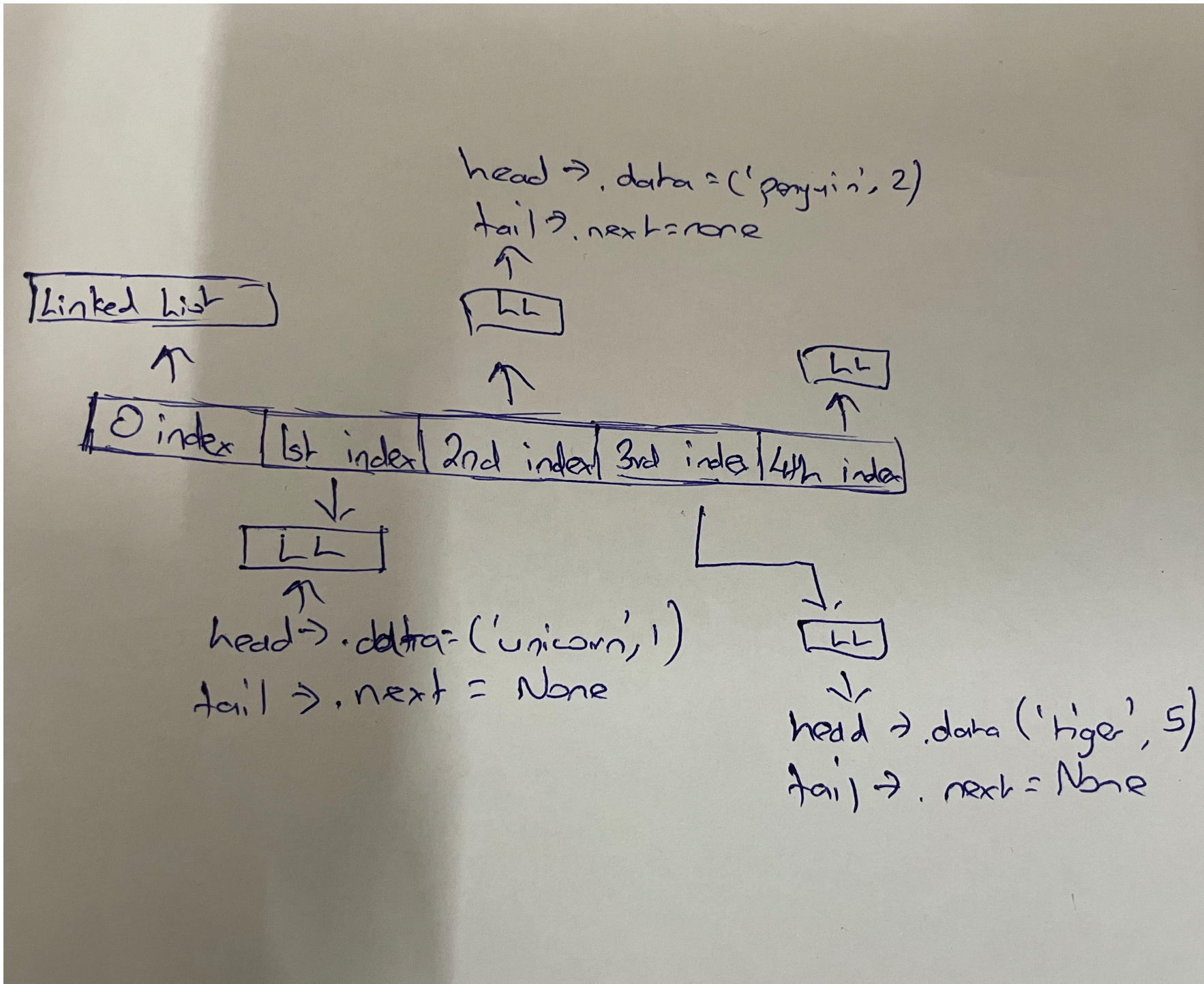
1. Call the hash function on the entry's key and then use the modulus operator (%) with the number of buckets to calculate the index of the bucket the entry belongs in.
2. Get the linked list the entry belongs in at this index in the array of buckets.
3. Add the entry's key and value to this list using its append operation.

Q3: What are the steps required to **retrieve an entry by its key** and **return its value**?

1. Get the bucket where the key and value is located
2. Find the key and return the value based on the key
3. If found return message saying that value found
4. If not found return error explaining key is incorrecct

Q4: Draw a diagram of **how a hash table data structure is organized in memory**. It contains the **4 key-value entries** listed below, has exactly **`b=5` buckets** and each bucket is a **linked list**. Label the buckets, their indexes and contents in appropriate places to complete the diagram.

key	hash(key)	value
'tiger'	393	5
'penguin'	642	22
'zebra'	273	8
'unicorn'	821	1


head → .data = ('penguin', 2)
tail → .next = none
Linked List
LL
LL
0 index
1st index
2nd index
3rd inde
4th inde
LL
LL
head → .data = ('unicorn', 1)
tail → .next = None
LL
head → .data ('tiger', 5)
tail → .next = None