CS 1.2: Intro to Data Structures & Algorithms

Hash Table Worksheet

Name: Mark Frazier

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 bucket the entry belongs in at this index in the array of buckets.
3. Add the entry's key and value to this bucket using its append operation.

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

1. Call the hash function on the key to find the index of the bucket where the entry should be.
2. Access the bucket at the recently found index in the array of buckets.
3. Iterate through the entries in the bucket to find the one with the matching key.
4. Once found, return the associated value.

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

Diagram on following page.

- Call the hash function on the hash(key) to find where entry should be

hash(key)	Index/Bucket	Key Value
273	0	(zebra, 8)
821	1	(unicorn, 1)
	2	
393	3	(tiger, 5)
642	4	(penguin, 22)

- Each bucket is represented as a linked list, with the contents listed horizontally.
- The buckets are indexed from 0 to 4.
- The entries are placed in their respective buckets based on the result of hashing their keys.