Megan Cole

CS-215

Assignment 5.1 - Hash Maps - Part 1

Part 1

A. A hash function is used to transform an input into a fixed-size numerical value called a hash code. The hash code is then used to determine the index within an array where the corresponding value will be stored or retrieved in the code.

B. A string hash function might be written by going through each character of the code, getting its Unicode value, and using math to generate a unique number. First, you have to initialize the hash to zero for a starting point. Second, we have to loop through each character of the string to get the current characters and the unicode value of each character. Third, we have to multiply the current hash by 31 and then add the unicode value of the characters. Finally, we are able to print our final hash code. Here is an example of how one could be written in Java:

a.

```
public int hashString(String key) {
                //Step 1
                int hash = 0;
                 // Step 2
                for (int i = 0; i < key.length(); i++) {
                        char currentChar = key.charAt(i);
                        intunicodeValue = (int) currentChar;
                // Step 3
```

```
                    hash = 31 * hash + unicodeValue;
                   //Step 4
                   return hash;
} //end hashString
```

C. We might use a hash function because it is faster than searching for a key when we search, add an entry, or delete an entry. When we search for a key, we have to go through the whole list of keys. However, when we use a hash function, it takes us directly to the location of where the key-value pair is being stored.

D. The Java Util HashMap function is based on the string's characters for hashing strings. The function uses polynomial rolling hash, which creates the hash code by multiplying a previous hash code by 31 and adding the value of the current character. This helps reduce the chance of having two strings with the same hash.