

Problem Set 9

Hash Table

1. You are given the following keys to be hashed into a hash table of size 11:

96, 43, 72, 68, 63, 28

Assume the following hash function is used

$H(\text{key}) = \text{key} \bmod 11$

and chaining (array of linked lists) is used to resolve collisions.

1. Show the hash table that results after all the keys are inserted.
 2. Compute the average number of comparisons for successful search.
-
2. Using chaining to resolve collisions, give the worst-case running time (big O) for inserting n keys into an initially empty hash table for each of the following kinds of chains:
- Chain is an unordered list
 - Chain is an ordered list (entries stored in ascending order of keys)
 - Chain is an AVL tree (ordered by keys)
-
3. Using the following class definitions:

```
class Node {
    int key;
    String value;
    Node next;
}

class Hashtable {
    Node[] entries;
    int numvalues;
    float max_load_factor;
    public Hashtable(float max_load_factor) { ... } // constructor
}
```

Complete the following methods of the Hashtable class, using the hash function $h(\text{key}) = \text{key} \bmod \text{table_size}$.

```
public void insert(int key, String value) {
    // COMPLETE THIS METHOD
}

private void rehash() {
    // COMPLETE THIS METHOD
}
```

Note: When expanding the hash table, double its size.

4. * Suppose you are asked to write a program to count the frequency of occurrence of each word in a document. Describe how you would implement your program using:
1. A hash table to store words and their frequencies.
 2. An AVL tree to store words and their frequencies.
- For each of these implementations:
1. What would be the worst case time to populate the data structure with all the words and their frequencies?
 2. What would be the worst case time to look up the frequency of a word?
 3. What would be the worst case time to print all words and their frequencies, in alphabetical order of the words?
- Assume there are n distinct words in the document, and a total of m words, and m is much greater than n .