

## CSCI124/MCS9124 Applied Programming    Spring 2014

### Assignment 4 Tips

#### An Example Scenario

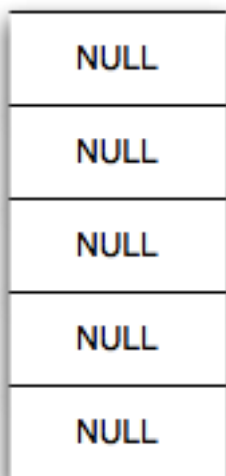
In this example I am going to show you how the chained hash table is built for the assignment. As you know the hash table is merely a collection of pointers to linked list nodes – each node typically holds some data i.e. the word/ key along with its frequency. Initially all elements of the hash table are pointing to NULL. In this example the hash table is five elements wide.

```
const int MAX = 5;

struct node
{
    . . .
    node* next;
};

node* hashtable[MAX] = {NULL};
```

As a consequence the hash table looks like this:



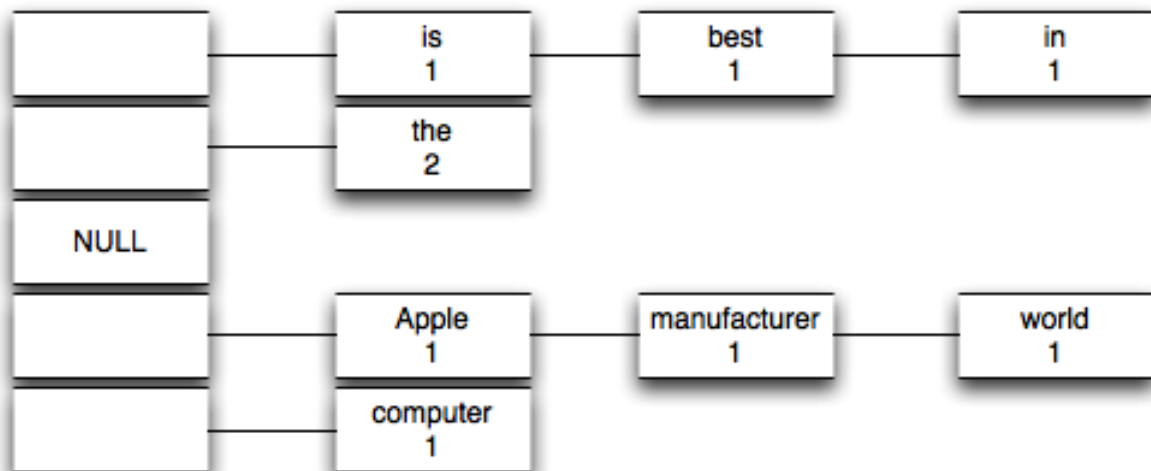
Consider the scenario whereby a text file contains the following sentence:

Apple is the best computer manufacturer in the world.

As per assignment specification this would result in the words (tokens) each having a combined ASCII value once going through the hash function You can also see what the hash index will be, by taking the combined ASCII value and % it against the hash table size (5).

Word	ASCII	HASH
Apple	498	3
is	220	0
the	321	1
best	430	0
computer	879	4
manufacturer	1293	3
in	215	0
the	321	1
world.	598	3

If each word is inserted in to hash table in the order stated above, the hash table would look as follows:



You can see that the hash table elements now point to words which are found in the collection. You already know that while populating a hash table we have to handle collisions - so how many collisions occur on the above input stream

Insertion of	Apple	yields no collision.
	is	yields no collision.
	the	yields no collision.
	<b>best</b>	<b>yields a collision on hash element 0.</b>
	computer	yields no collision.
	<b>manufacturer</b>	<b>yields a collision on hash element 3.</b>
	<b>in</b>	<b>yields a collision on hash element 0.</b>
	the	does not yield a collision because it exists in the chained list.
	<b>world.</b>	<b>yields a collision on hash element 3.</b>

A **collision** occurs when the hash element is pointing to a list and the word being inserted does not exist in the linked collection. Remember a collision **ONLY** occurs **WHEN INSERTING NEW** data into the hash table. As this is a chained table we need to consider the list. Given the above scenario you **ONLY** have 4 collisions. A better way to think about this is when each of the words/ token maps to a unique hash table entry there are **NO collisions**.