CS 310
Assignment 304
March 2, 2022

Jingbo Wang

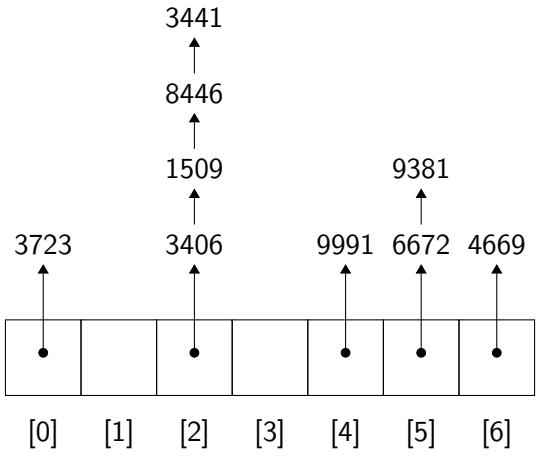
Problem 1 — Given the input 3406, 6672, 4669, 1509, 8446, 9991, 3723, 9381, and 3441, in that order, and the hash function $h(x) = 6 - (x \bmod 7)$, draw the resulting hash table. State and explain any assumptions you make.

Answer:

Using hash function $h(x) = 6 - (x \bmod 7)$, we get:

$$h(3406) = 6 - (3406 \bmod 7) = 6 - 4 = 2$$
$$h(6672) = 6 - (6672 \bmod 7) = 6 - 1 = 5$$
$$h(4669) = 6 - (4669 \bmod 7) = 6 - 0 = 6$$
$$h(1509) = 6 - (1509 \bmod 7) = 6 - 4 = 2$$
$$h(8446) = 6 - (8446 \bmod 7) = 6 - 4 = 2$$
$$h(9991) = 6 - (9991 \bmod 7) = 6 - 2 = 4$$
$$h(3723) = 6 - (3723 \bmod 7) = 6 - 6 = 0$$
$$h(9381) = 6 - (9381 \bmod 7) = 6 - 1 = 5$$
$$h(3441) = 6 - (3441 \bmod 7) = 6 - 4 = 2$$

Here is the hash table:



Problem 2 — Assume that conditions indicate a need to rehash the hash table from problem 1. Perform a rehash of that hash table, and show the hash table that results from the rehashing. Explain your process and results.

Answer:

Because the table gets almost full, and has too much open chaining, we need to rehash the original hash table.

First, we need to allocate a new table of size being more than twice as big as the current table, and also size is prime number. $17 > 2 \times 7$

Therefore, we have new hash function: $h(x) = x \bmod 17$.

Then, we have:

$$h(3406) = 3406 \bmod 17 = 6$$

$$h(6672) = 6672 \bmod 17 = 8$$

$$h(4669) = 4669 \bmod 17 = 11$$

$$h(1509) = 1509 \bmod 17 = 13$$

$$h(8446) = 8446 \bmod 17 = 14$$

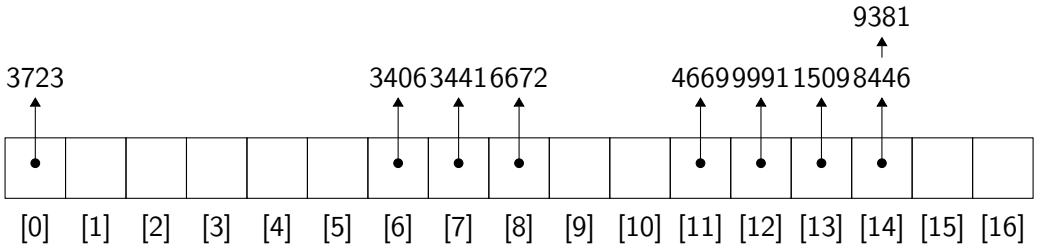
$$h(9991) = 9991 \bmod 17 = 12$$

$$h(3723) = 3723 \bmod 17 = 0$$

$$h(9381) = 9381 \bmod 17 = 14$$

$$h(3441) = 3441 \bmod 17 = 7$$

There is the hash table:



Problem 3 — Every standard Unix system has a file /usr/share/dict/ words which is a newline-delimited utf-8 file with many common words and names. The command
$ wc -l /usr/share/dict/words will tell you how many lines are in the file (the wc command stands for wordcount; here, there is one word per line).

Answer:

```
const unsigned TABLE_SIZE = 102409;
string word;
size_t hash_num = 0;
unsigned collisions_sum = 0;
vector<unsigned> hash_array(TABLE_SIZE, 0);
while (getline(cin, word))
{
  hash_num = hashx(word, TABLE_SIZE);
  hash_array.at(hash_num)++;
  if (hash_array.at(hash_num) > 1)
  {
    collisions_sum++;
  }
}
cout << "The number of collisions: " << collisions_sum << endl;
```

Using the command

`$ wc -l /usr/share/dict/words,`

we can get lines in the file are: 102401.

Running my program with the command line,

`$ cat /usr/share/dict/words | ./program`

we can get the result is: 37748.

Problem 4 — State, explain, and justify the results you got from running the program in question 3. Be sure to explain what table size you used in your program, and why you chose that value, and what the load factor used by your program is.

Answer:

For the table size in the Line 1, I use 102409 instead of 102401 form Using the command,

```
$ cat /usr/share/dict/words | ./program
```

because 102401 is not a prime number, and 102409 is smallest prime number large than 102401.

The load factor is:

$$\begin{aligned}\frac{n}{m} &= \frac{102401}{102409} \\ &= 0.9999218\end{aligned}$$

It is close to 1, so `TABLE_SIZE` = 102409 is correct.

In the Line 5, I create a vector named `hash_array` initializing the vector to 0 to count the number of collisions the algorithm produces when the words are hashed.

For example, if `hash_num` in Line 8 equals to 10102, and `hash_array.at(10102)` equals zero and runs Line 9: `hash_array.at(10102)++;` for one time, then it means that 10102 appeared 1 time in
`/usr/share/dict/words`. And if 10102 appeared second times, it will run Line 9 again, then it means that 10102 appeared 2 times.

For Line 10 – Line 13, if a number appeared more than 1 times, it will produce one open chaining, so `collisions_sum++`.

Thus, we can get the result is: `collisions_sum` = 37748 in Line 15.

Also, we can know the time complexity, Line 6 always runs when read the `word` file for n times:

$$T(n) \in \Theta(n)$$