Word Unscrambler Design Process

Riley Tallman

CSE 310

Professor Nakamura

Spring 2018

Adventure Works: The ultimate source for outdoor equipment

Original Design:

For every permutation:

1. Read every line of a text file dictionary
2. Compare with the current permutation
3. Print it if it was a match

Permutations: $n!$

Permutations*235969 = number of comparisons

Big-Theta of $n!$

Next Design: Hash Table with chaining

- Decide average linked list iteration: 8
  - Make 8 comparisons on average to see if a permutation matches anything in the dictionary
- Searching time for one permutation:$\theta(8) = \theta(1)$
- Searching time for all permutations: $\theta(1) * \theta(n!) = \theta(n!)$
- Big-Theta is the same, but its much faster than previous because we do not have to compare 235,969 words. We only compare 8 on average.

Hash Table with chaining

- 235,969 words / 8 average linked list = 29,495.125
- Closest power of 2 = 32,768
- Closest prime number = 29,501
- Size of table = 29,501
- Actual average searching time: 235,969 / 29,501 = 7.998 comparisons

Initialization

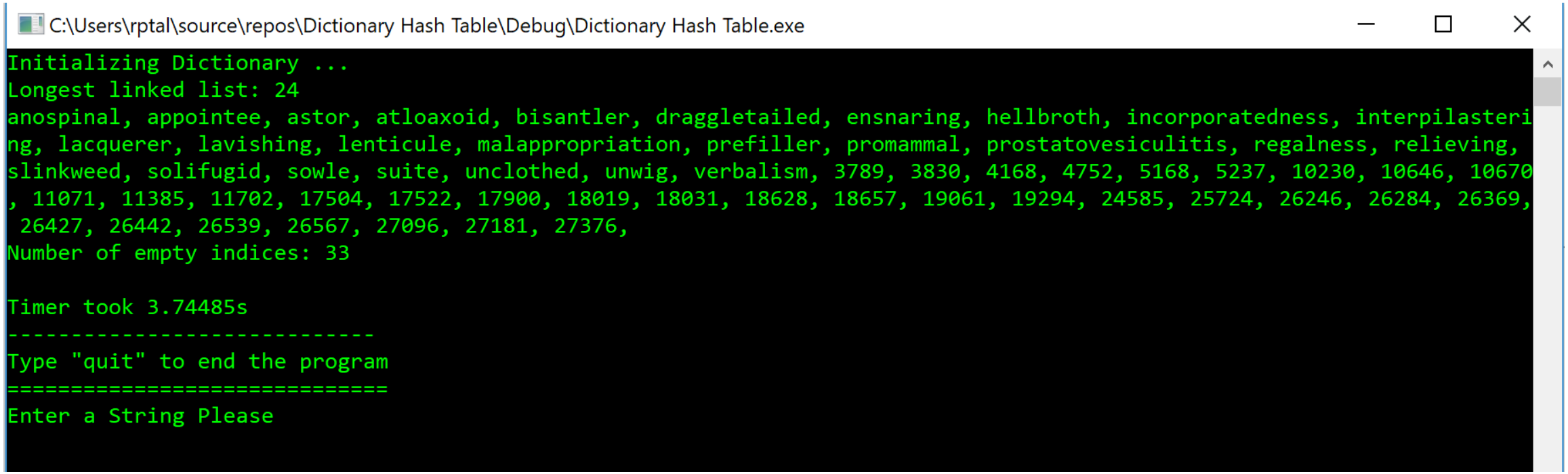
- Create dictionary array of linked list with the size of the dictionary
- Read text file with dictionary words
- The word is the key
  - Hash the key
  - Insert into array at the end of the linked list

Choosing Good Hash Function

1. Large enough to achieve highest index location (29,500)
2. Variety in operations
   Need more than just add and subtract
3. Use as many variables as possible
   Using constants will result in the same values for the same letters

```
int hashh(string key)
{
    int index = 1;
    for (int i = 0; i < key.length(); i++)
        index += (key.at(i))*(i + 65) + (key.at(i)%(i+1));

    index = abs(index);     // its negative if overflowed
    return index %= dSize;      // index mod dSize
}
```



Longest linked list = 24

Empty indices = 33 --> means good average

Took less than 4 seconds to initialize

```
Enter a String Please
a

All words with letters "a" will now be listed!
a,

There were 1 matching words :)

Average Linked List: 1
Timer took 0.0027367s
----------------------------
Enter a String Please
no

All words with letters "no" will now be listed!
no, on,

There were 2 matching words :)

Average Linked List: 5.5
Timer took 0.00606285s
----------------------------
Enter a String Please
yes

All words with letters "yes" will now be listed!
sey, sye, yes,

There were 3 matching words :)

Average Linked List: 7.83333
Timer took 0.0314792s
```

- 1,2,3 Letter words

```
Enter a String Please
then

All words with letters "then" will now be listed!
hent, neth, then,

There were 3 matching words :)

Average Linked List: 11.9583
Timer took 0.00326215s
----------------------------
Enter a String Please
today

All words with letters "today" will now be listed!
toady, today,

There were 2 matching words :)

Average Linked List: 12.05
Timer took 0.0142474s
----------------------------
Enter a String Please
eating

All words with letters "eating" will now be listed!
eating, ingate, tangie,

There were 3 matching words :)

Average Linked List: 8.2375
Timer took 0.0228786s
```

- 4,5,6 Letter words

```
Enter a String Please
toasted

All words with letters "toasted" will now be listed!

There were no matching words :(

Average Linked List: 10.9321
Timer took 0.0498505s
----------------------------
Enter a String Please
tomorrow

All words with letters "tomorrow" will now be listed!
moorwort, rootworm, tomorrow, wormroot,

There were 4 matching words :)

Average Linked List: 6.05238
Timer took 0.0582415s
----------------------------
Enter a String Please
chocolate

All words with letters "chocolate" will now be listed!
chocolate,

There were 1 matching words :)

Average Linked List: 9.39457
Timer took 1.64239s
```

- 7,8,9 Letter words

```
Enter a String Please
everything

All words with letters "everything" will now be listed!
everything,

There were 1 matching words :)

Average Linked List: 7.96158
Timer took 40.744s
-----------------------------
Enter a String Please
everything

All words with letters "everything" will now be listed!
everything,

There were 1 matching words :)

Average Linked List: 7.96158
Timer took 31.3672s
-----------------------------
Enter a String Please
characters

All words with letters "characters" will now be listed!

There were no matching words :(

Average Linked List: 10.9006
Timer took 10.4327s
```

- 10 Letter words

```
Enter a String Please
advertising

All words with letters "advertising" will now be listed!
advertising,

There were 1 matching words :)

Average Linked List: 8.51338
Timer took 358.468s
-----------------------------
Enter a String Please
personality

All words with letters "personality" will now be listed!
personality,

There were 1 matching words :)

Average Linked List: 5.29178
Timer took 673.179s
```

- 11 Letter words
- Takes a looong time

Improvements

- Reduce average searching time from 8 to 4 or something.
  - Uses more memory
- Alphabetized Linked List?
  - If the key is less than the head of the linked list then do not check the rest of the list
- Check key.length()?
  - If the length of the key does not equal the word then do not compare the individual letters.
- Always have n! permutations of the letters -> $\theta(n!)$

Thank you

Riley Tallman