

⑤ continued calculate(a, b) returns the total runtime for loading the dictionary

⑥ The Freed loop scans the document entered as an argument and Sends words are sent at a time into the spellchecker function, checking to make sure its not really numbers

Load load dictionary into memory, i.e.

word\n
word\n
word\n
word
word

Pseudocode

1. open dictionary
2. read from dictionary into buffer
3. pass buffer through hash function
4. malloc node at hash index
e.g. if hash is 3
malloc node here
5. set char value at malloced node to word in buffer
6. Go to step 2

Pseudocode is incomplete — forgot to account for the process of adding to a linked list

HASH

Example input

- Food SKY Apple Life

N = 143,091 words in Large dict.

| | |
|---|-----|
| a | 97 |
| e | 101 |
| i | 105 |
| o | 111 |
| u | 112 |
| q | 122 |

Food

$$\begin{array}{r}
 \text{Ascii} \quad 102 \quad 111 \quad 111 \quad 100 \\
 \begin{array}{r}
 429 \\
 \times 1002 \\
 \hline
 1111 \\
 \times 111 \\
 \hline
 10,800 \\
 \hline
 12,321
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 (102 + 100) - \\
 202 - 222 \\
 (111 + 111)
 \end{array}$$

$$\frac{1155}{2} - 20$$

↓

0

$$\begin{array}{r}
 9 = 4.5 \\
 2 \text{ true} \\
 4 \\
 \text{ round } 5
 \end{array}$$

forgetful

0.34 string-1

Check Search for char in hash table

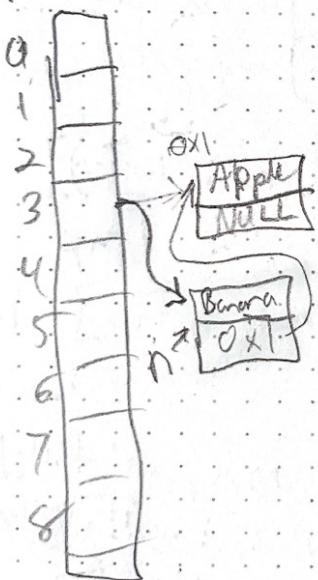
Pseudo Code

1. Produce Hash associated with given word
e.g. Apple \rightarrow 3253

2. Go to index 3253 in hash table

3.

Review Linked Lists Apple + Banana



int *
linked
list

Add to
linked
list

malloc node at 3
Apple + Banana

table[3] becomes
head

table[3] \rightarrow word = Apple
table[3] \rightarrow next = NULL

malloc node at n
n \rightarrow word = Banana
n \rightarrow next = table[3]

table[3] = n

This will affect
our Load Function

Check (Revisiting Load)

initializing $\text{table}[N] \rightarrow \text{next} \rightarrow \text{NULL}$

for ($\text{int } i = 0; i < N; i++$)

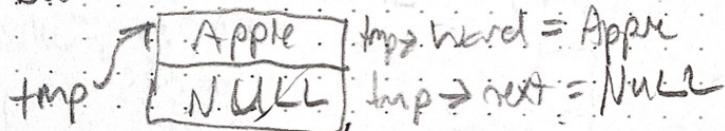
$\text{table}[i] \rightarrow \text{next} = \text{NULL}$



1. We've read the word
into a buffer

Control 1 → if $\text{table}[\text{hash}(\text{buff})] == \text{NULL}$

2. malloc tmp node containing
buffer (say, apple). Init to NULL



3. Have $\text{table}[\text{hash}(\text{buffer})]$
(say, 3) point to tmp

$\text{table}[\text{hash}(\text{buffer})] = \text{tmp}$

Control 2 → else

2. malloc tmp node, init $\text{table}[\text{hash}]$
 $\text{tmp} \rightarrow \text{word} = \text{init Value}$

$\text{tmp} \rightarrow \text{next} = \text{table}[\text{hash}]$

3. $\text{table}[\text{hash}] = \text{tmp}$

to (set next value)

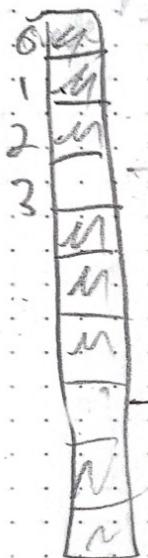
updateHash(tmp)

Check (Revisiting Load)

Add a trav pointer that keeps track of most recent node

Add 4. To 1st control

4. trav = table[hash(buff)]



Add 4. 2nd control

4. trav = tmp
address of trav is now
tmp

OK Now we need
trav doesn't
work b/c each index
needs a trav for hash
How do you make it work?

I was right before, we need to
add nodes at beginning of list