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

See pset5.pdf for these notes

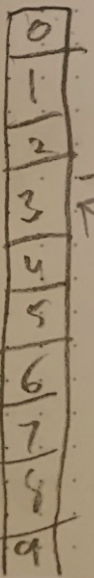
(6) The Fread loop scans the document entered as an argument and sends words one at a time into the spellchecker function, checking to make sure it's not reading numbers

Load load dictionary into memory, i.e. into

Pseudocode

word \n
word \n
word \n
word
word

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



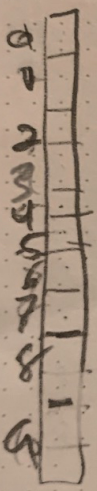
Pseudocode is incomplete — forgot to account for the process of adding to a linked list

HASH

Example input

- Food Sky Apple Life

lv = 143,091 words in Large dict.



Food

Ascii 102 111 111 100

102 × 100

111 × 114

10,200

12,321

a 97
e 101
i 105
o 111
u 112

(102 + 100) − 202 − 222

(111 + 111)

$\frac{11}{2}$ 5.5 −20 → 0

a-z

97-122

forgetful

0 3.4 strlen−1

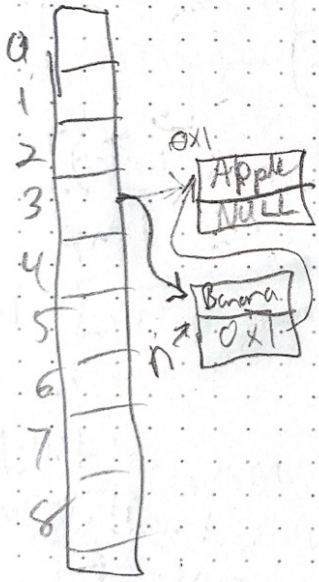
$\frac{9}{2}$ = 4.5 trunc 4 round 5

Check search for char* in hash table

Psuedo code

1. Produce a Hash associated with given word e.g. Apple → 3253
2. Go to index 3253 in hash table
3. ~~~~~~~~~~~~~~~~~~~~

Review Linked Lists



Apple & Banana

malloc node at 3

Initialize linked list

table[3] becomes head

table[3]→word = Apple

table[3]→next = NULL

Add to linked list

malloc node at n

n→word = Banana

n→next = table[3]

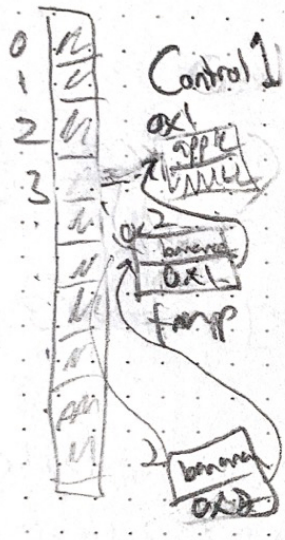
table[3] = n

This will affect our Load function

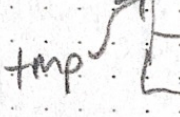
Check (Revisiting Load)

Initializing table[N] → next to NULL

```
for (int i = 0; i < N; i++)
    table[i] → next = NULL
```



1. We've read the word into a buffer
   if table[hash(buff)] == NULL
2. malloc tmp node containing buffer (say, apple). Init to NULL



3. Have table[hash(buffer)] (say, 3) point to tmp

   table[hash(buffer)] = tmp

Control 2 → else

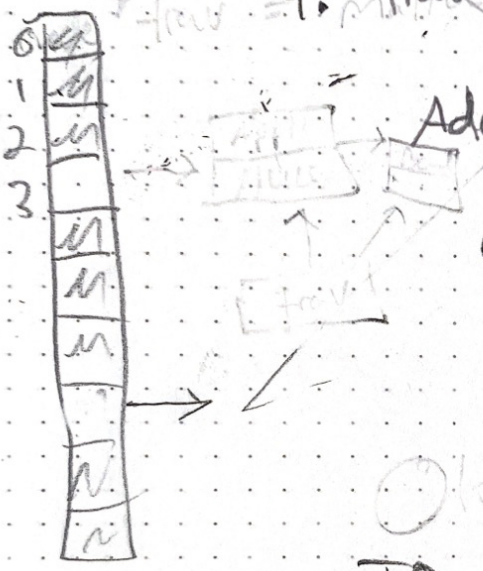
2. malloc tmp node, init table[hash]
   tmp→word = banana
   init Value
   tmp→next = table[hash]
3. table[hash] = tmp

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

If trav doesn't work b/c each we'd need a trav for every index

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