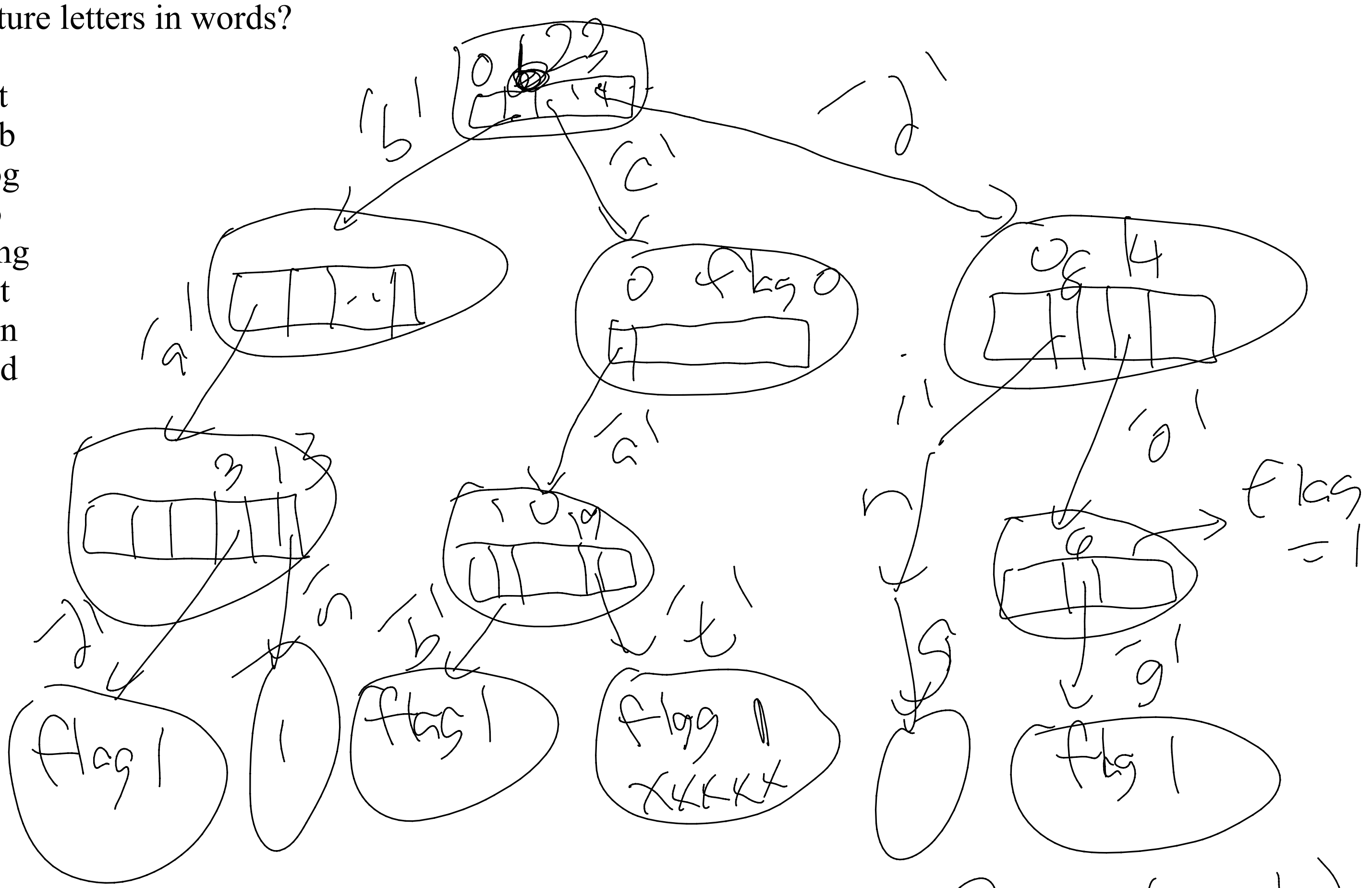
7/1/2020 - Tries

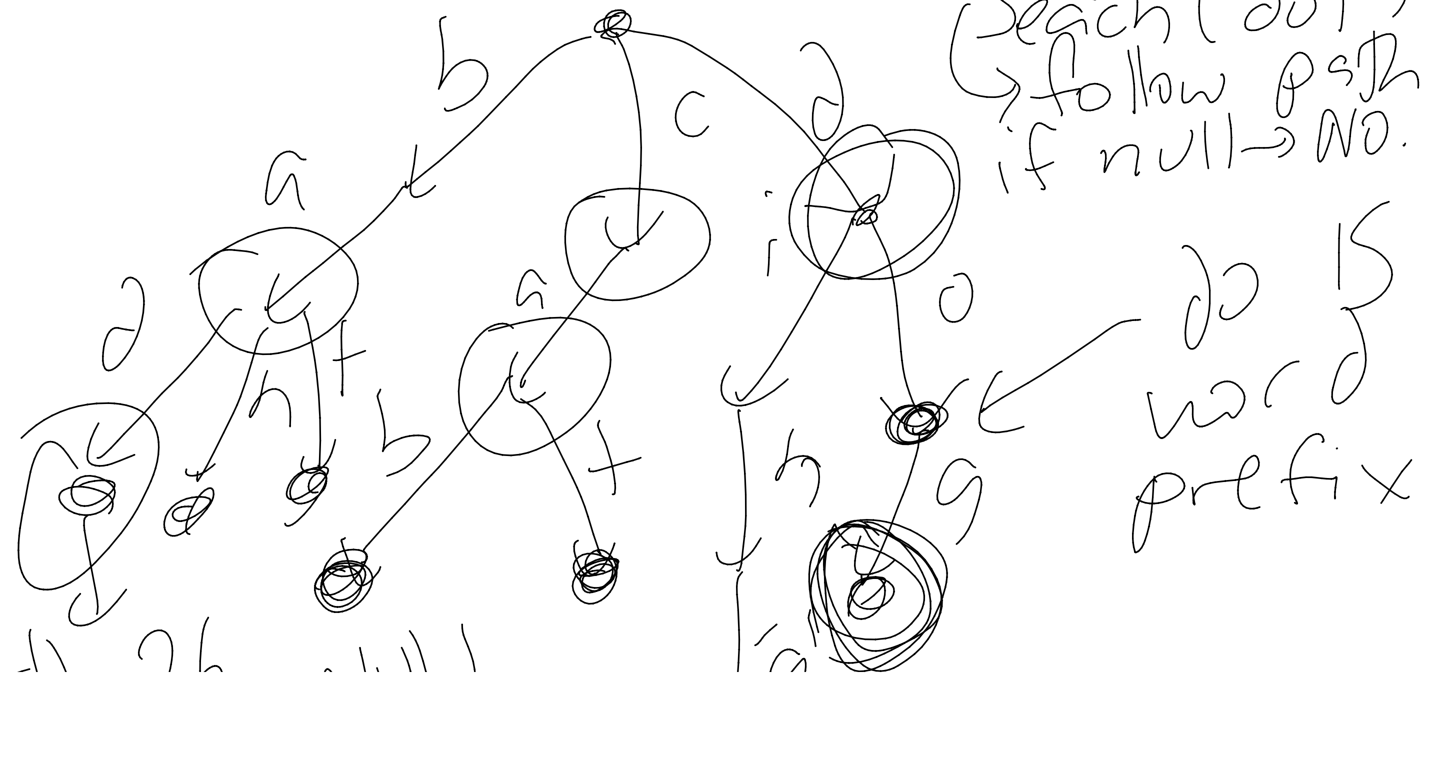
Wednesday, July 1, 2020 4:04 PM

A regular binary tree has 2 pointers to subtrees, left and right.

For English, we have 26 letters, so what if we stored 26 pointers, to future letters in words?

cat
cab
dog
do
ding
bat
ban
bad





all 26
nexts = NULL

```
typedef struct trienode {
    int flag;
    struct trienode* next[26];
} trienode;
```

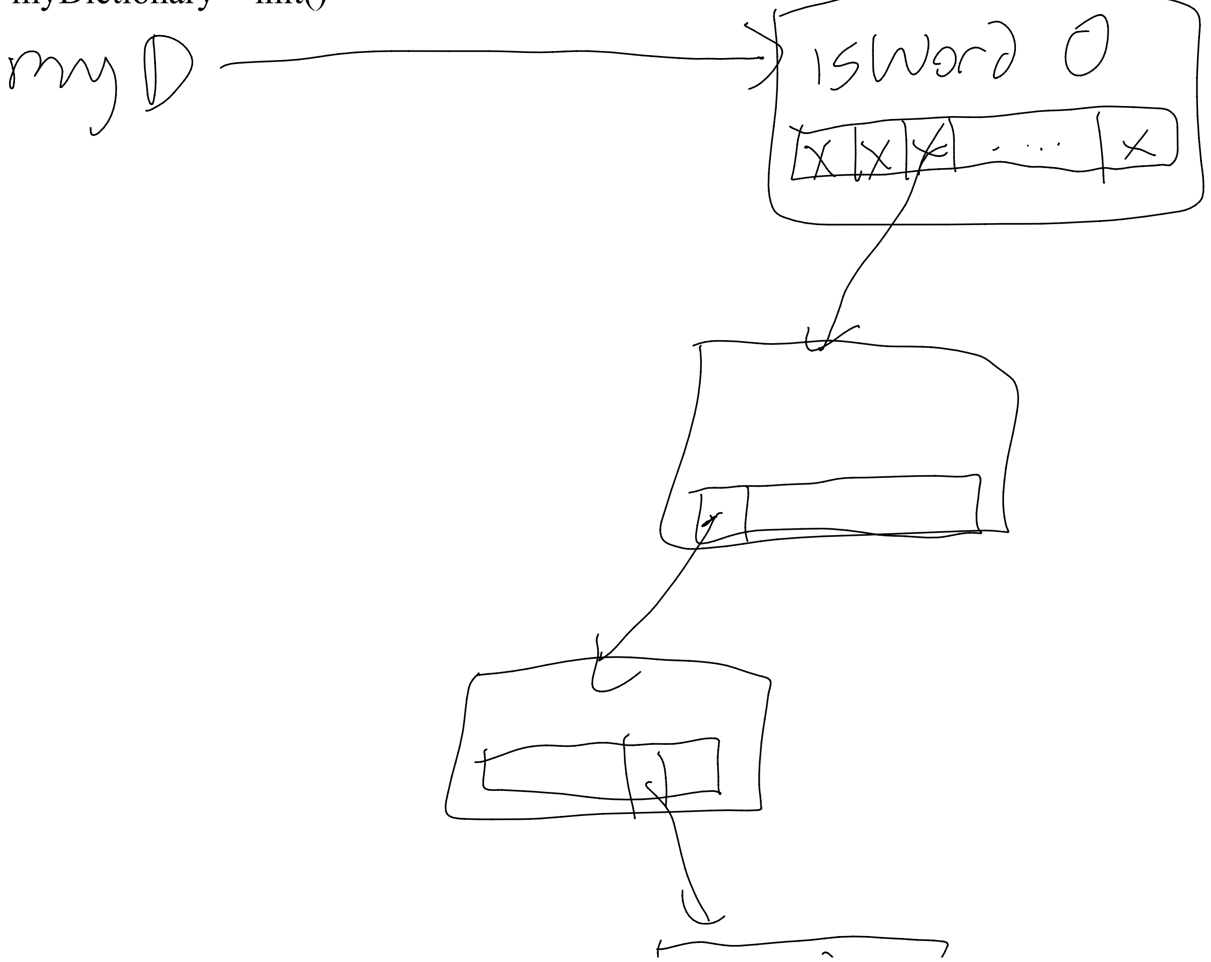
array size 26

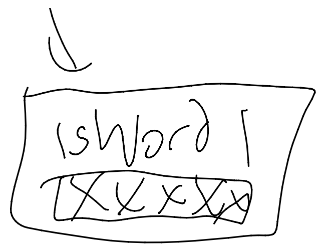
array of pointers!

index to the array is a letter, essentially…actually it's the equivalent integer from 0 to 25 where a->0, b->1, c->2 …, z->25

word[i] - 'a' (is how we convert to the 0 to 25 range)

main
------
myDictionary = init()





search is like insert…follow the path down. if you get stuck (ie trying to go down a null ptr, return 0, word isn't there.) otherwise, just return the flag when you get to the appropriate end of the word.

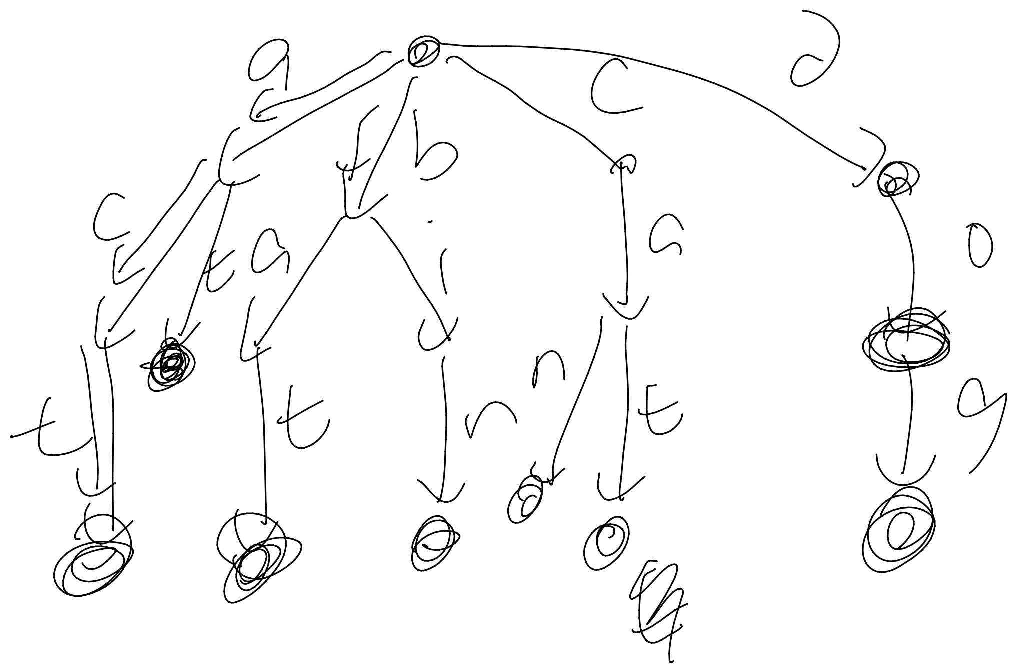
run time of a search and insert = O(n), where the word has n letters.

Typical user case…

Dictionary 150,000

binary search log 150,000 comparisons, about 17 or so.

In a trie, the average run time would be the average length of a word, which is probably about 5 or 6 at most…if you get rid of words that "the" that hopefully people don't search for, then maybe this average could be a bit bigger, but probably not that much. (Avg 8.41)



act
at
ant,

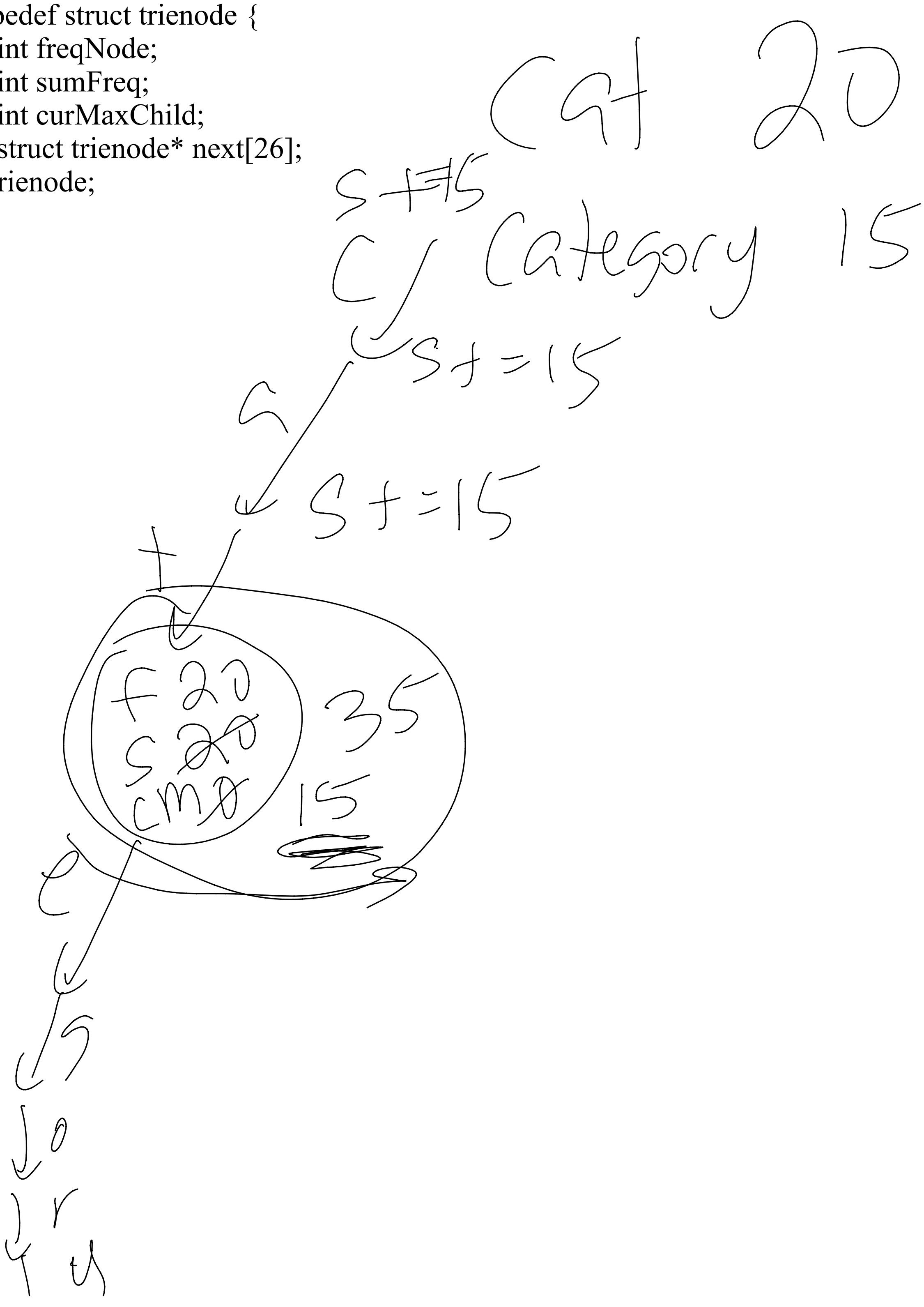
To print all, we'll do a traversal and pass in a buffer that keeps track of where we are, word wise. When we get to a node that is a word, we will null terminate our buffer and print.

is a word, we will null terminate our buffer and print.

dat
bin

node for P4:

```
typedef struct trienode {
    int freqNode;
    int sumFreq;
    int curMaxChild;
    struct trienode* next[26];
} trienode;
```



We looked at three problems:

1. Return the # of words with a particular prefix (given that the total # of words in a trie is stored in the node.)
2. Return the number of words that can be formed with a set of given Scrabble tiles.
3. Find a word such that a maximum number of its prefixes are also words…just find the value of that maximum number (not the word itself.)