

COP 3502 11/5/25 (call mom!)

1) Don't come to class Fri (11/7/25), but watch Hash Table Video on main Webcourses page. (1 hr 20 min)

2) Will enter Com Service today!

3) Post Assn Study Group Notes (1 turn in) (10 pts total) → Min 3 mtgs outside rec.
→ TAs are allowed to change groups.

3.5) P5 is fixed! Due later 11/16. P6 Due 12/1

*4) Start Tries

MONDAY - Some Tries, Heap

Tries

Goal: Store a dictionary of words so we can look up if a word is in the dictionary or not (insert, delete, search). Best use case: lots of words but max length is pretty short.

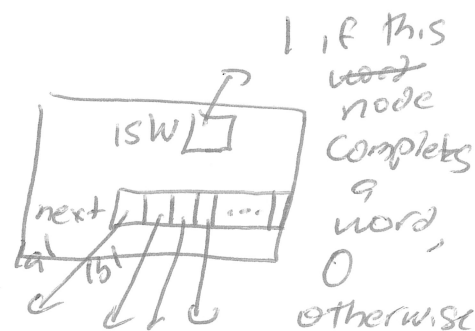
26-ary tree

struct trie {

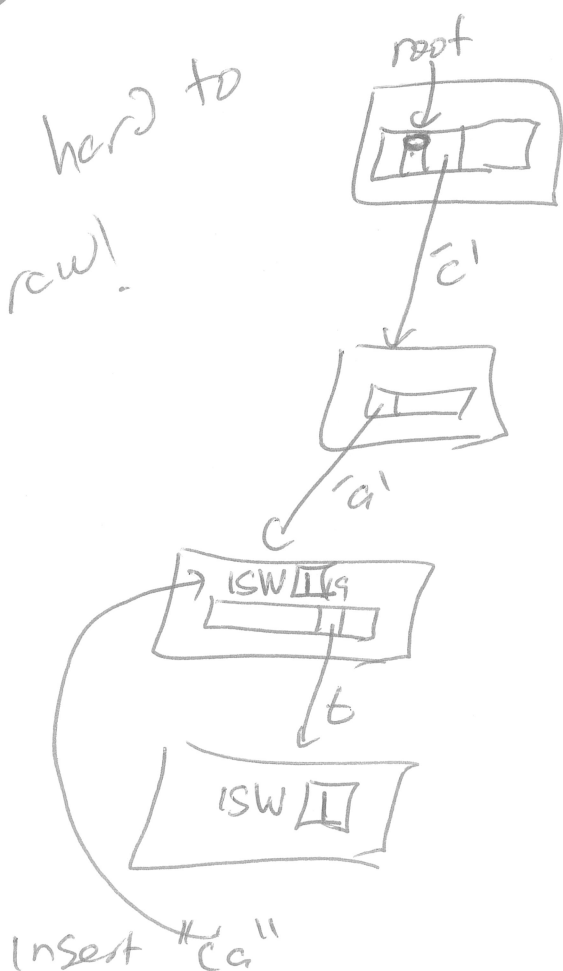
int isWord;

struct trie* next[26];

};



too hard to draw!

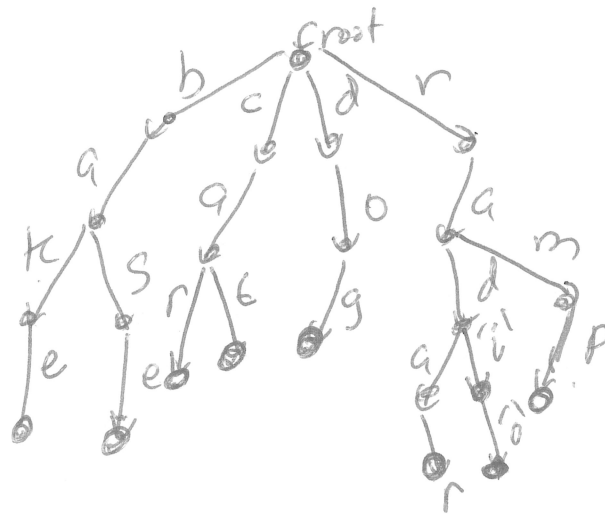


insert cat
search caption
pseudocode sketch
of search

```
Search(trie* root, char* str) {
    int len = strlen(str);
    for (int i = 0; i < len; i++) {
        if (root == NULL)
            return 0;
        root = root->next[str[i] - 'a'];
    }
    if (root == NULL)
        return root != NULL &&
        root->isWord;
}
```

Visual drawing

Insert cat
dog
radar
ramp
base
batce
~~act~~
car
Insert "radio"



1) insert $\rightarrow$ follow path like search I wrote, BUT if a node doesn't exist malloc it. When you get to end set its $isWord = 1$.

2) delete - ^{run} search code to traverse to node, then change $isWord = 0$.