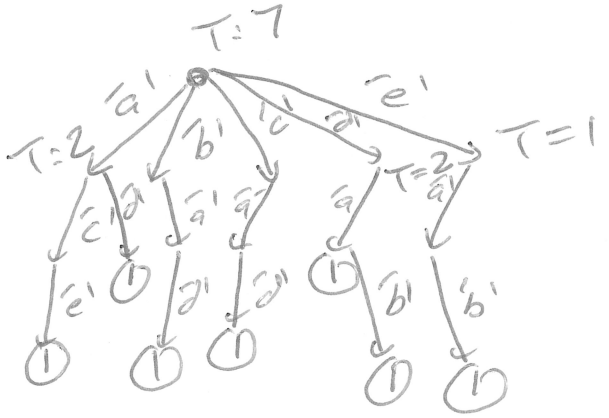
COP 3502 3/7/2024

Tries - Dictionaries of "words"

best use case: all words are relatively short

most runtimes based on either word length (height) or # words ($\sim$ # nodes x ~~length of word~~ $\sim$ #words x length word)

list
- acev
- dav
- adv
- cadv
- badv
- ~~be~~
- eabv
- dab ✓



No letters are stored anywhere. We can determine the letters by looking at which array index we "travel" on.

(char)('a' + i)

not always necessary

143423
-140294
3129

```
struct trienode {
    int isword;
    int total; // # words in this subtrie
    struct trienode* next[26];
};
```

(a) read in a dictionary

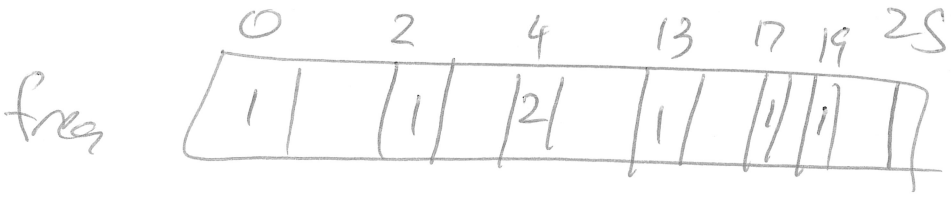
(b) printing all words

(c) answer queries about # words that start w/a given prefix.

(d) Scrabble Words
C, A, E, E, N, R, T

To search for suffixes, reverse words before inserting and ~~reve~~ reverse the suffix when searching.

Scrabble



C A
E E
N R T

```
go( root , freq, str, k )
{
    if  isword
        print

    for (i=0; i<26; i++) {
        if (ptr == NULL) continue;
        if ( freq[i]==0 ) continue;
        freq[i]--;
        go (root, freq, str, k+1);
        freq[i]++;
    }
```

east nl ~~t~~ r, h, d, i, c

10^7 odometer

$8^7 = 2^{21} \approx 2$ million

~~$6 \times 9 \times 8 \times 7$~~

$\binom{10}{7}$

a c d e h i l n r s t