

Tries \Rightarrow Auto complete

Prefix trees \rightarrow n array tree.

Ex1 : [Yash, Yah, Had, Hame, Pock, Poc]

Ex2 : [1101, 1100, 1000, 10001]

Ex3 : IP addresses.

Ex4 : Phonenumbers.

Ex5 : College Id

Trie

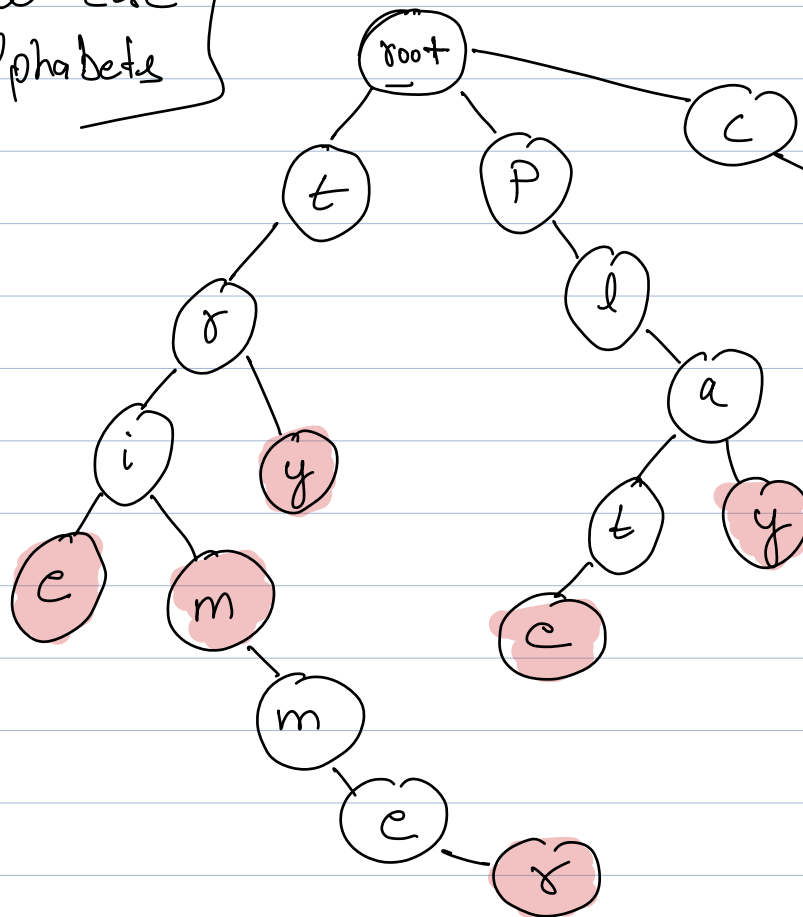


Trie, Try, Trim, Play.
Plate, CAR, TRIMMER

[small case
alphabets]

n, l

Tri



+rim

class Node {

char data;
Node child[26];
bool isEnd;

Node left;
Node right;

Sc: (n * l * 26)

}

Insertion !

void insert (Node root, String word) {

Node curr \Rightarrow root;

int len \Rightarrow word.size();

0 - 25

for (int i = 0; i < len; i++) {

int ind \Rightarrow word[i] - 'a';

if (curr.child[ind] == null) {

curr.child[ind] = new Node(word[i]);

}

curr = curr.child[ind];
}

curr.isEnd = true;

}

Sc: $O(1)$

Tc: $O(l)$

\downarrow
length of the word.

Search \Rightarrow search if a word is Present

bool search (Node root, string word) {

Node curr \Rightarrow root;

int len \Rightarrow word.size();

for (int i = 0; i < len; i++) {

int ind \Rightarrow word[i] - 'a';

if (curr->child[ind] == null) {

return false;

}

curr = curr->child[ind];

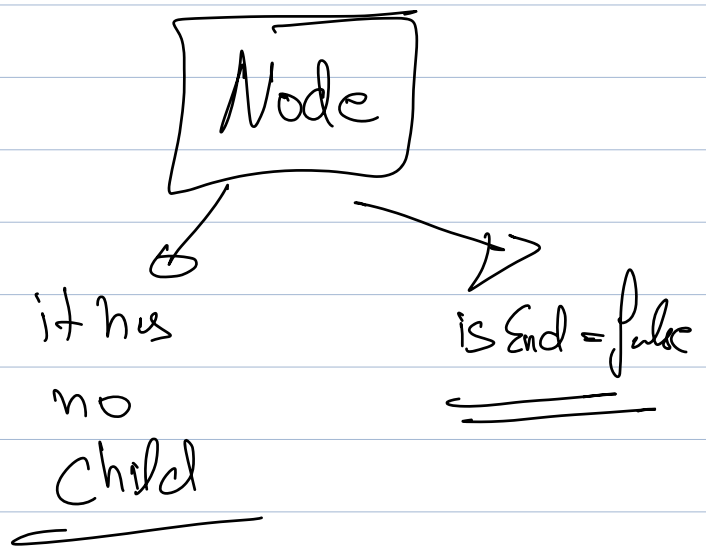
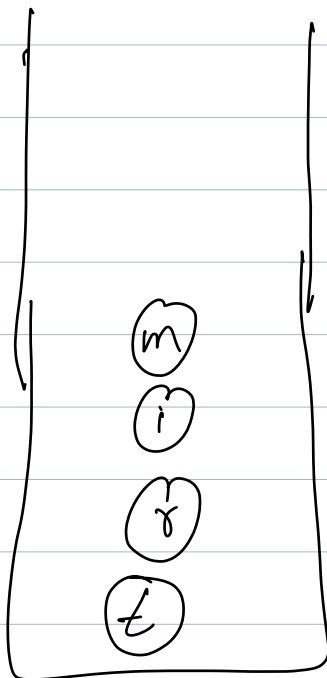
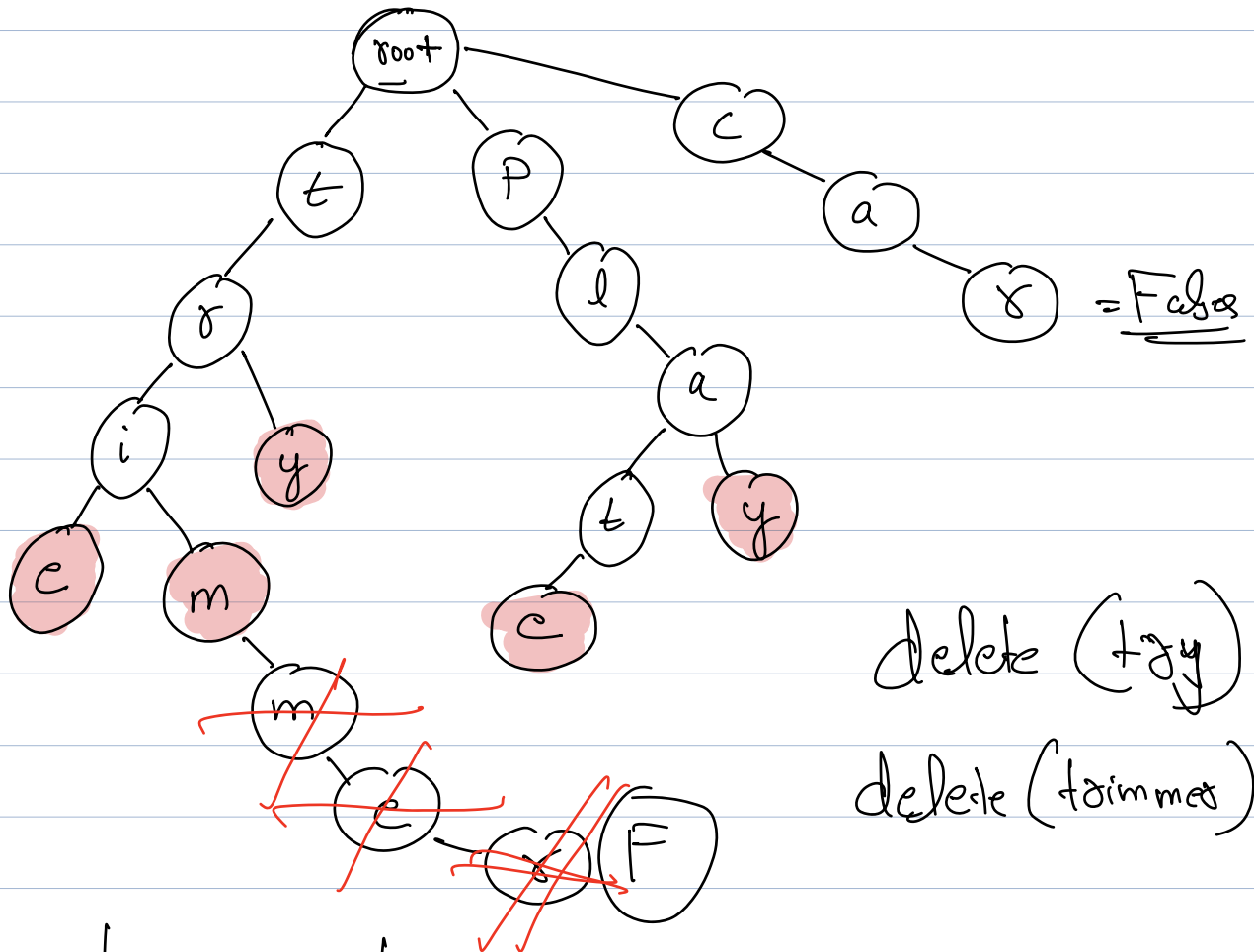
}

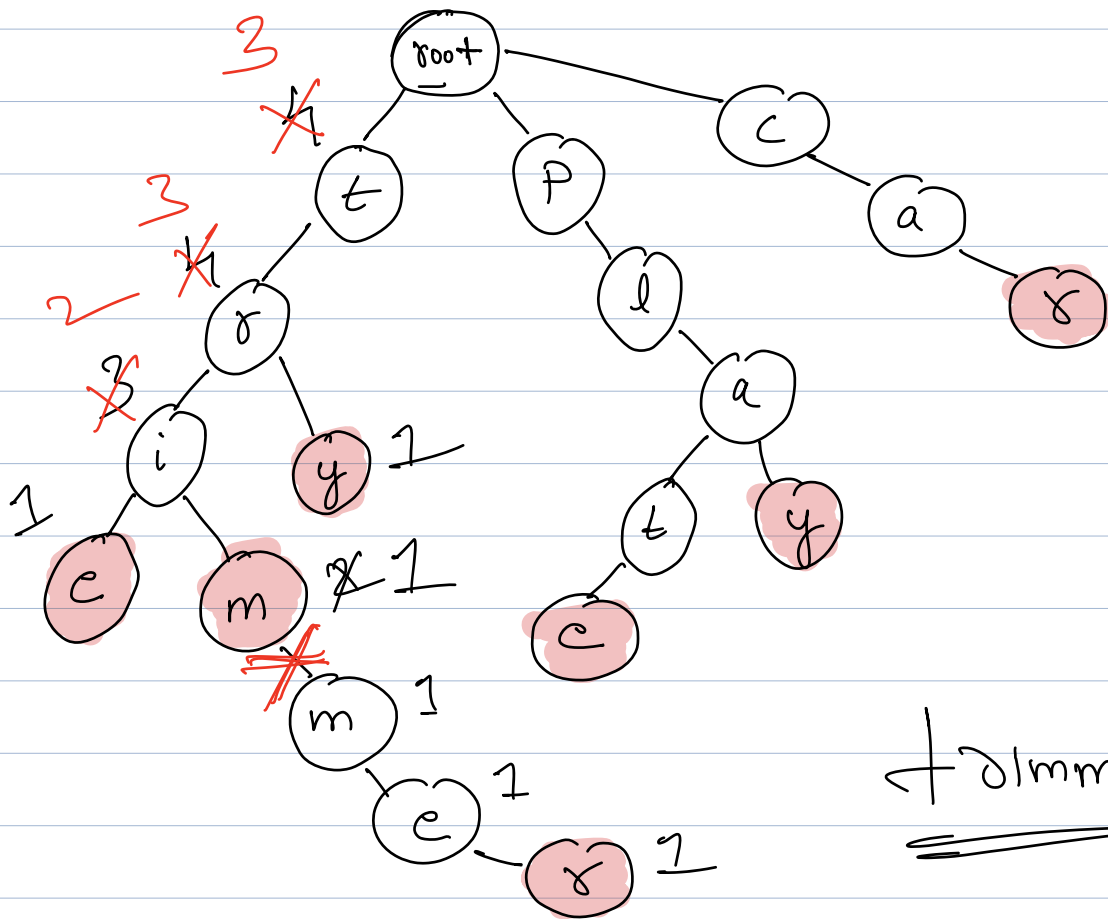
return curr->isEnd;

}

Deletions

delete (case)



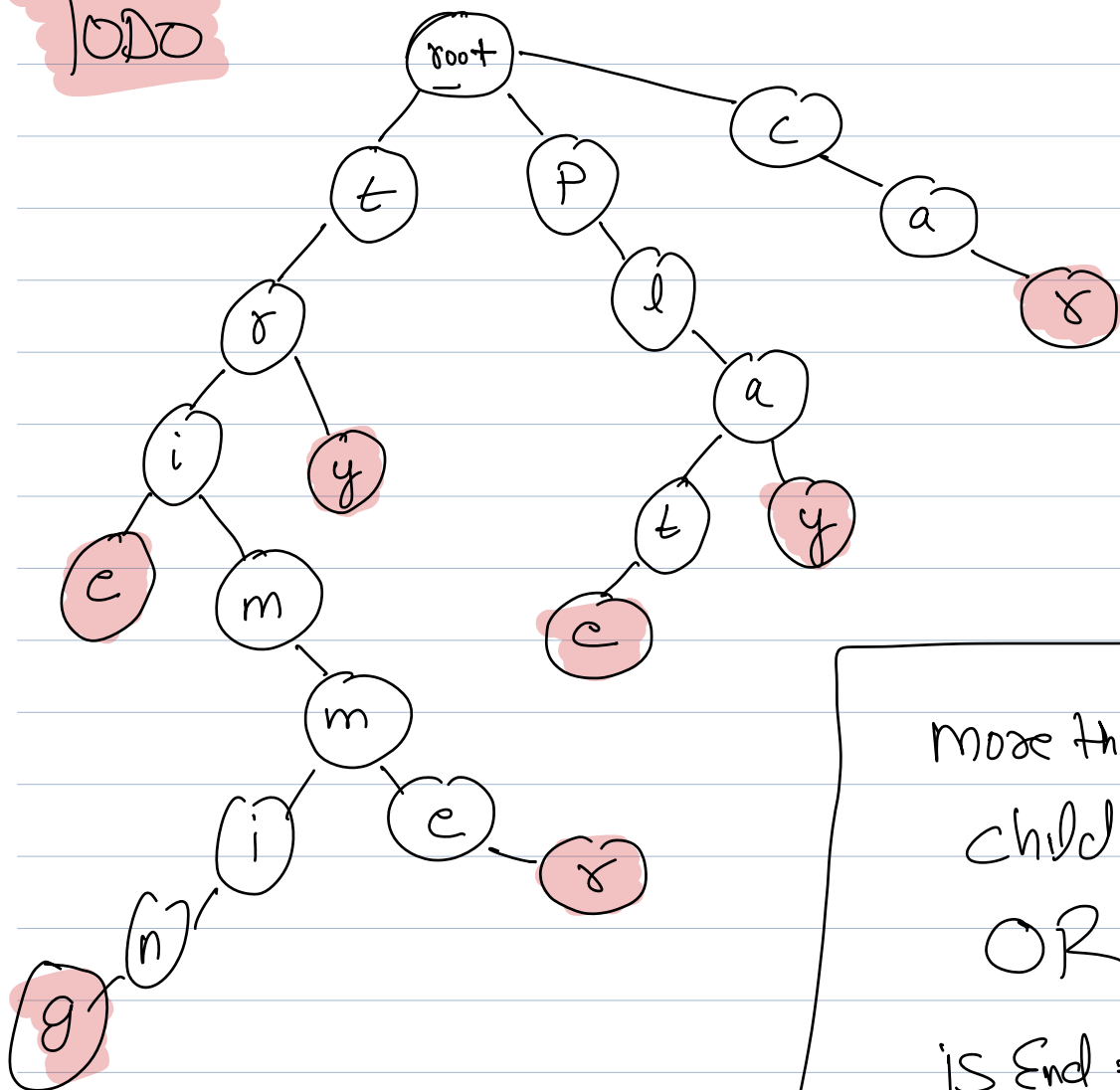


dimmer

curr \Rightarrow m

trimming

Todo



more than 1

child

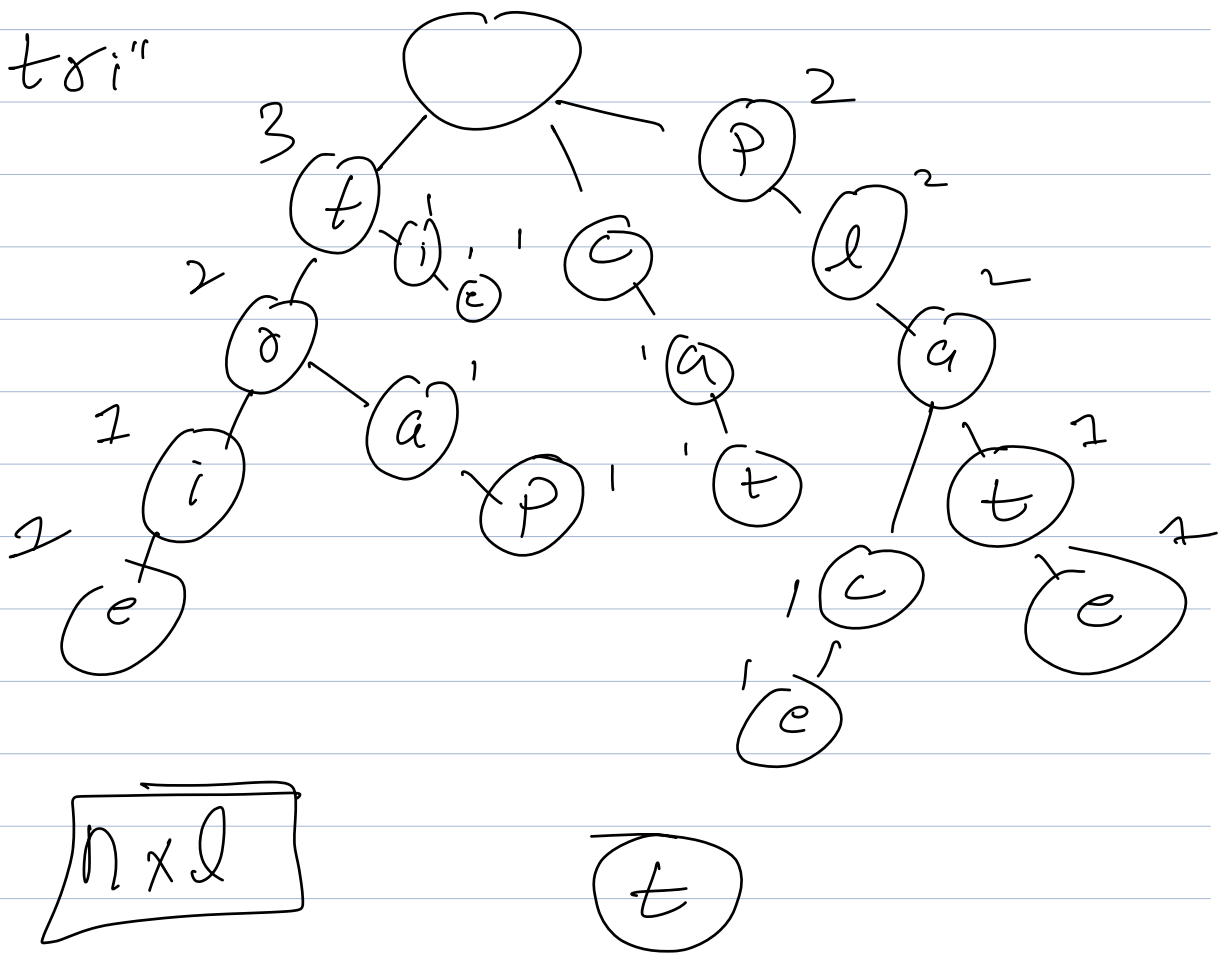
OR

is End = true

Q Find shortest unique prefix to represent each word.

trie	trap	plate	cat	place	tie
t	t	p	c	p	t
tri	tra	plat	c	plac	ti

$S \rightarrow "tri"$



the dimming form.

map <string, int>

$t' \rightarrow t_{rie}$

$(t_r' \rightarrow t_{rie}$

$(t_{ri}' = dr$

$'t_{rie}' = dr.$

$t \rightarrow H_{isc}$

$H \rightarrow \mathbb{Z}$

tree

→ more than
1 child

→ is End = True.

⏏
not for the
word we
are searching

