Implementation of Trie

Thursday, 25 November 2021 7:56 PM

Need of Linked list.

→ Insertion is easy, in terms of complexity.

→ Deletion is easy .. ..

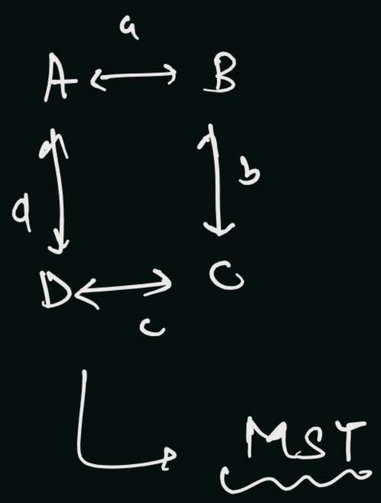
→ Non-continuous Memory.

→ Allowed space: Until memory is non-empty.

Importance of Tree (Including all types)

→ Heirarchical D.S.

→ File System → Based on Heirarchy.



Importance of Graph

We can travel in Both direction.

→ Cyclic path can be solved using graph.

Requirements :→

Insert → "apple"

search → "apple" → True.

Search → "app" → False.

start with → "app" → True.

Insert - "app"

Search - app → True.

Insert → "cards"

Search → "car" → False.

start with → "car" → True.

start with → "carol" → True.

Search "Card" → False

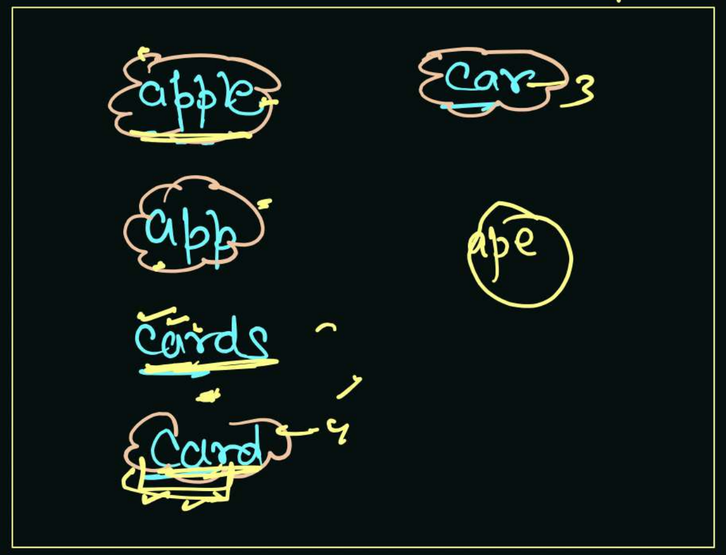
Insert "carol"

Insert "car"

start with "ca" → True.

Character → a → a_2 → p_2 → p_2 → l_1 → e_1 (trace)

p_2 → e_1



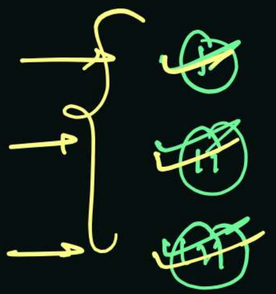
Data base

Start with

cal

functionalities requirements from TRIE :→

Software



Insert → add complete word in DB

Search → find exact word in DB

Startwith → self explanatory a substring start with 'a'

Traditional Methods of performing above operations →

Database → ArrayList <String>

Insert "apple"

Insert "app"

Search - "apple" → 4a3

"apple" | "app"

4e3

for start with - "app"

a → apple, app

b → battle, bat

for Insert and Search → Hashing

length vs string

5 → "apple"

3 → app

* Start with ??

using HashMap

character vs string

storing character — complete string

n! → check fn → n!

Searching

1. Brute force, travel on every string and check if it is equal to string at ith character.

Insert →

- ✓ CAR
- ✓ TRIE
- ✓ TRIED
- ✓ CARDS #
- ✓ TRY
- ✓ CAT
- ✓ COT
- ✓ CARS
- ✓ CARIED
- ✓ CRY
- ✓ TEMP
- ✓ GOOD

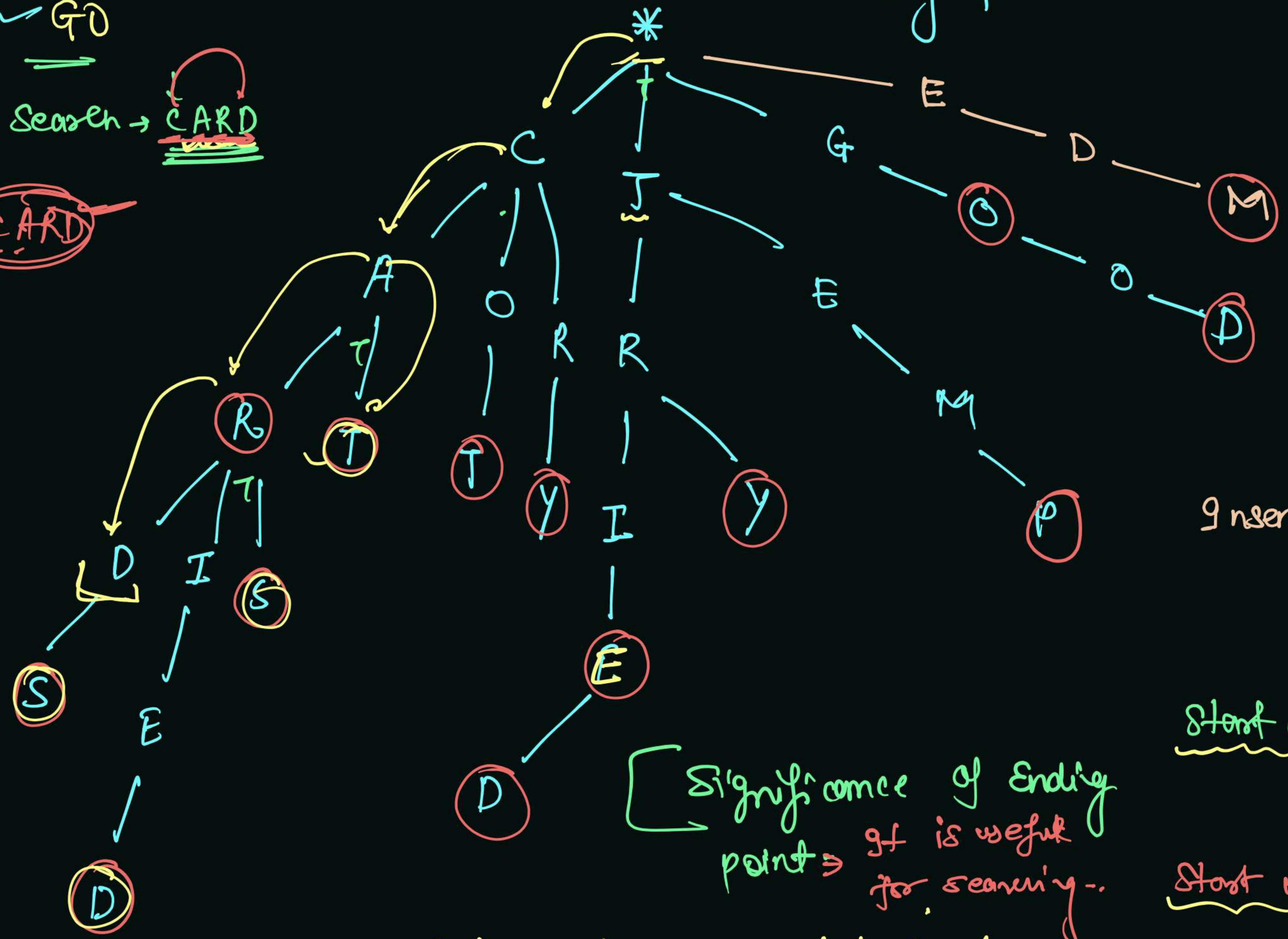
search CATTLE

✓ GO

Search → CARD

CARD

Root → starting point



Insert "EDM"

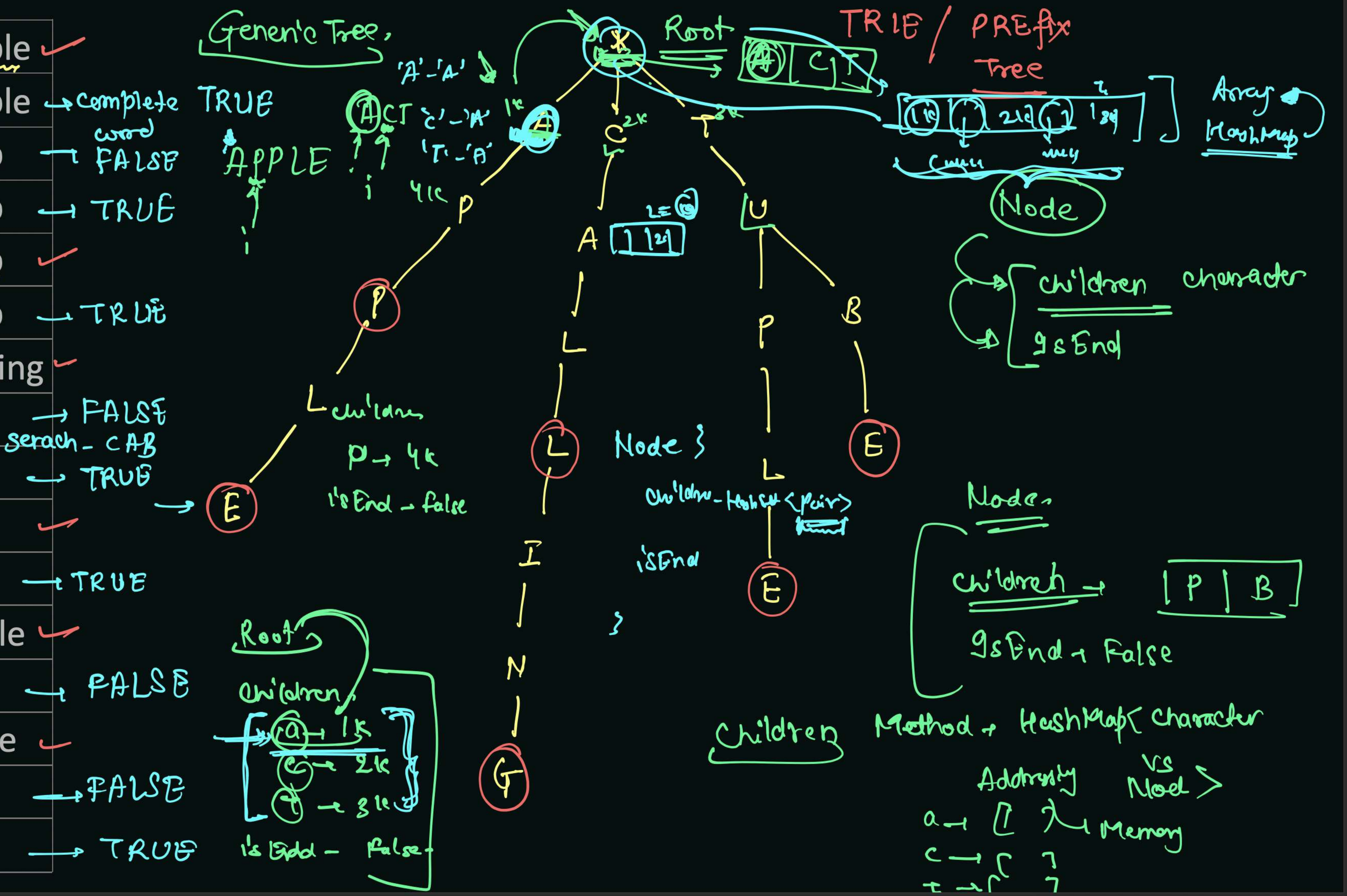
[Significance of Ending point ⇒ It is useful for searching..

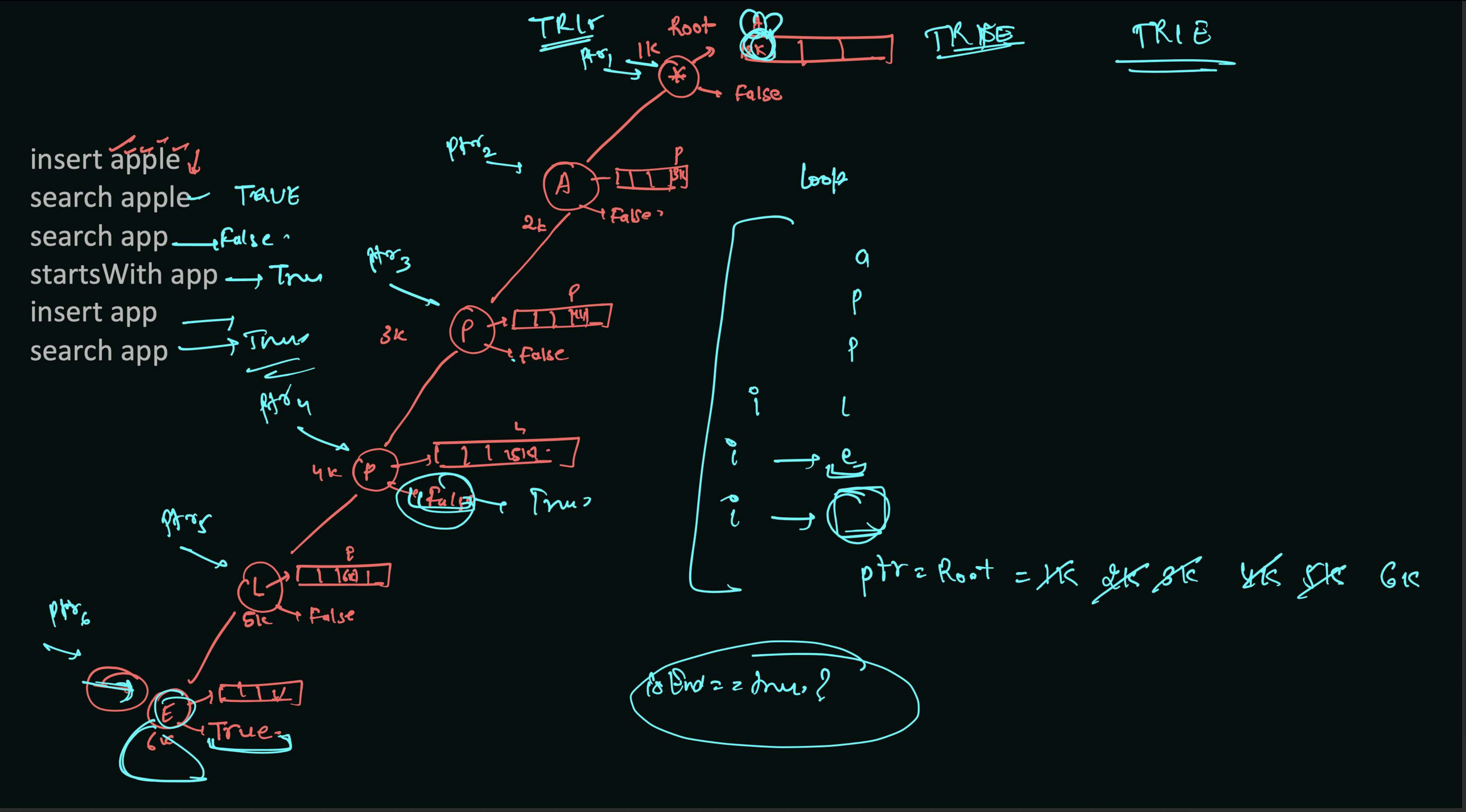
Endpoint marking is useful for "search".

Start with "TRIE"
└ TRUE

Start with "CA"
└ TRUE

insert	apple
search	apple
search	app
startsWith	app
insert	app
search	app
insert	calling
search	call
startWith	call
insert	call
search	call
insert	tuple
search	tub
insert	tube
search	tub
startWith	tub



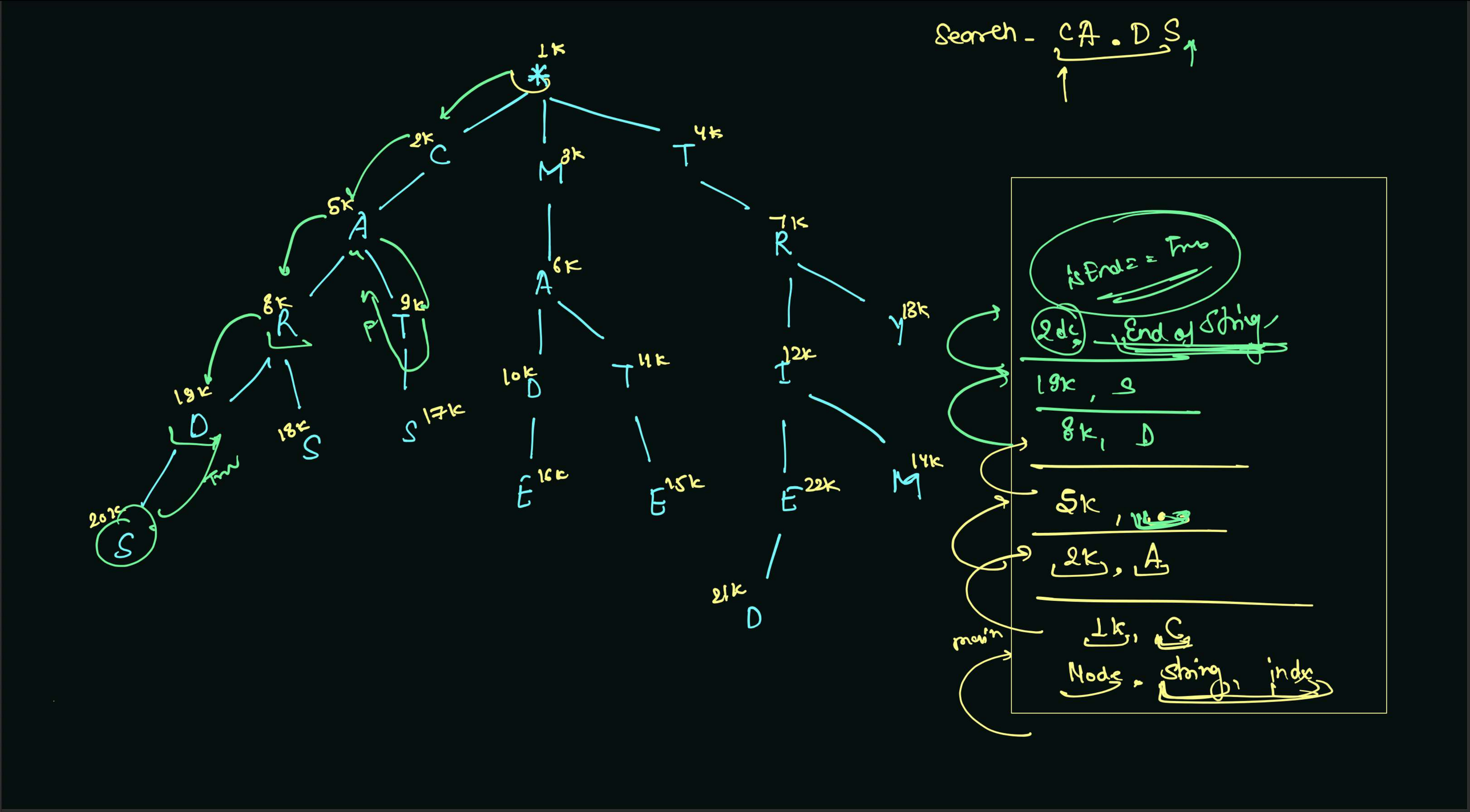

TRIE
TRIE
TRIE
Root
1K
ptr1
*
False
ptr2
A
2K
False
ptr3
P
3K
False
ptr4
4K
P
False
True
ptr5
L
5K
False
ptr6
E
6K
True
insert apple
search apple TRUE
search app False
startsWith app True
insert app
search app
True
loop
ptr = Root = 1K 2K 3K 4K 5K 6K
isEnd == true ?

Design Add and Search Word Data Structures

Thursday, 25 November 2021 7:58 PM

- Insert tuple
- Insert tried
- Insert tries
- Insert true
- Insert trump
- Search tr..s
- Search t...p
- Search ...d
- Insert tuned
- Search tu..d


```
if (word.charAt(i) == '.') {
    for(int i=0; i<26; i++)
    {
        if(recursion && u==true)
            return true;
    }
}
return false;
```


Search - CA . DS
1K
2K
C
M 3K
T 4K
5K
A
R 7K
A 6K
8K
R
9K
T
P
y 13K
19K
D
18K
S
S 17K
10K
D
T 11K
I 12K
20K
S
E 16K
E 15K
E 22K
M 14K
21K
D
isEnd = True
2dc
End of String
19K , S
8K , D
5K ,
2K , A
1K , C
main
Node , string , index