TRIE

- Insert

apple

apps

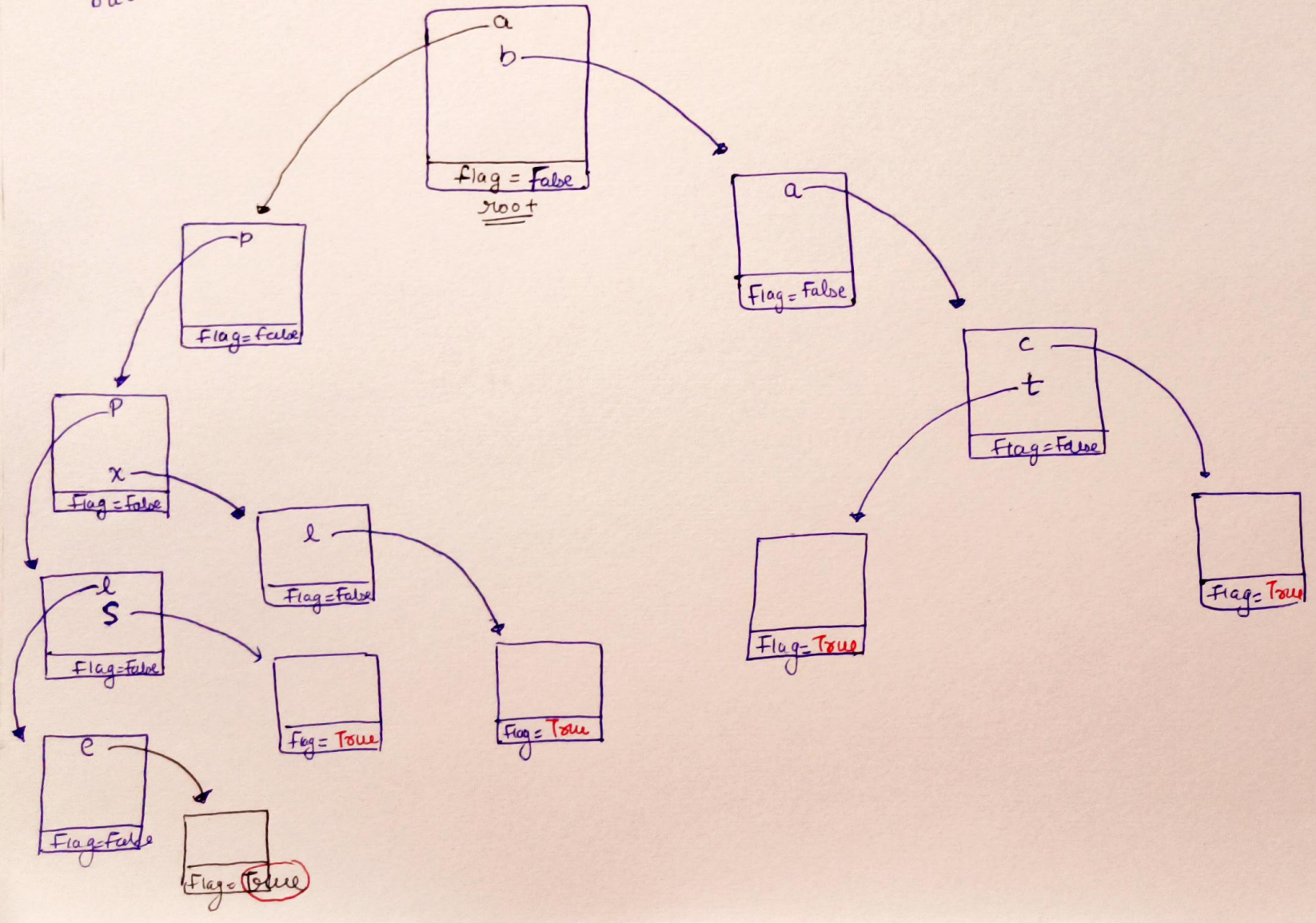
apxl

bac

bat

```
Node {
    Trie * a[26];
    bool flag;
}
```

0	1	2	.....	25
↓	↓	↓		↓
a	b	c		z



TRIE IMPLEMENTATION

```
class Node{
    public:
    Node* link[26];
    bool flag = false;

    bool contains(char ch)
    {
        return (link[ch-'a'] != NULL);
    }

    void put(char ch, Node* node)
    {
        link[ch-'a'] = node;
    }

    Node* get(char ch)
    {
        return link[ch-'a'];
    }

    void setEnd()
    {
        flag = false;
    }

    bool isEnd()
    {
        return flag;
    }
};
```

check if reference trie is present or not

creating a reference Trie

To get the next node for traversal

Setting Flag to true at the end of the word

checking if the word is completed or not.

```
Class Trie
{
    Node *root;

    Public:

    Trie(){
        root = new Node();
    }

    void insert(string word)
    {
        Node * node = root;  // Initialising dummy node pointing to root.
        for(int i=0 ; i<word.size(); i++)
        {
            if(!node->contains(word[i]))
            {
                node->put(word[i], new Node());
            }
            //Move to reference Trie
            node = node->get(word[i]);
        }

        node->setEnd();

    }
```

T.C. → O(len(word))

```
bool Search (string word)
{
    Node* node = root;
    for (int i=0; i<word.size(); i++)
    {
        if (!node->contains(word[i]))
        {
            return false;
        }
        node = node->get(word[i]); //→ Move to reference Trie
    }
    If (node->isEnd()) return true;
    return false;
}
```

T.C → O(length(word))

```
bool startsWith (string prefix)
{
    Node* node = root;
    for (i=0 to i<prefix.size())
    {
        if (!node->contains(prefix[i]))
        {
            return false;
        }
        node = node->get(prefix[i]);
    }
    return true;
}
```

T.C → O(length(word))