NCKU Programming Contest Training Course
Course 15
2015/05/20

Tzu-Yen Chiu(tommy5198)
tommy5198@*gmail.com*

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

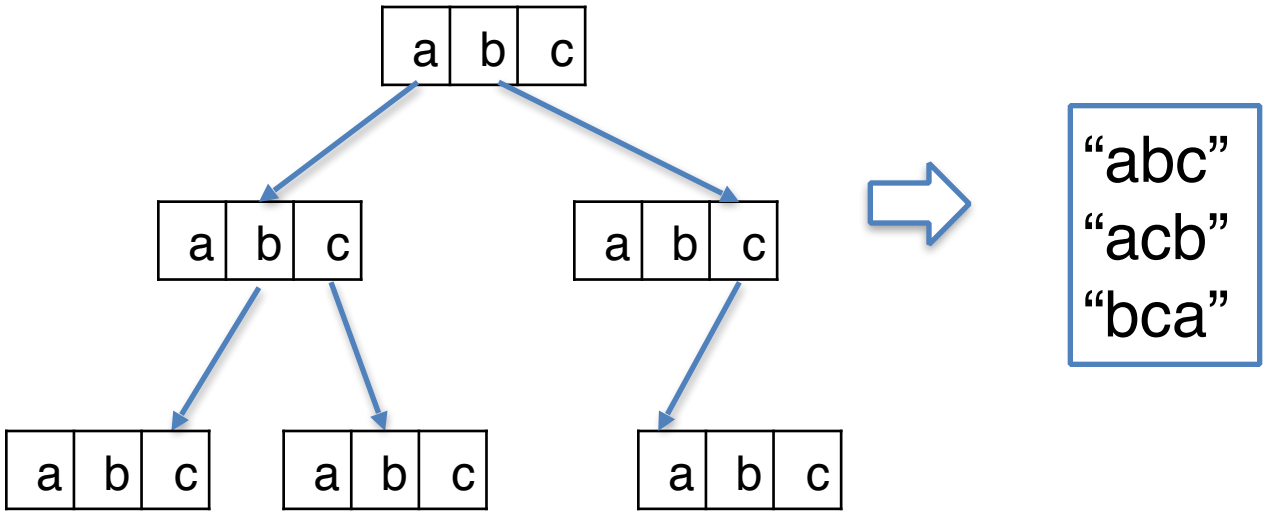
Outline

- Trie
  - String data structure
  - dictionary

made by tommy5198

Trie

- Store strings into tree structure



made by tommy5198

Trie

- Structure detail
  - Charater (128 for ASCII)
  - Counter
  - Other if needed

```
struct Trie{
    Trie* next[128];
    int cnt;
}*root;
```


made by tommy5198

Trie

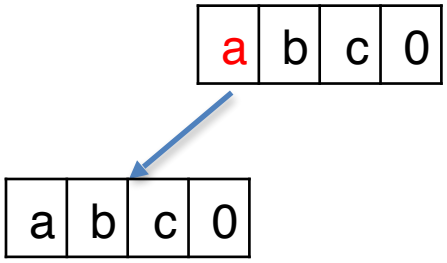
- Add string into trie
  - Create new node for each charater
  - Increase counter at the end

```
void add(char *str) {
    Trie *tmp = root;
    while(*str) {
        if(tmp->next[*str] == NULL)
            tmp->next[*str] = new Trie();
        tmp = tmp->next[*str];
    }
    tmp->cnt++;
}
```

Trie

- Add string into trie

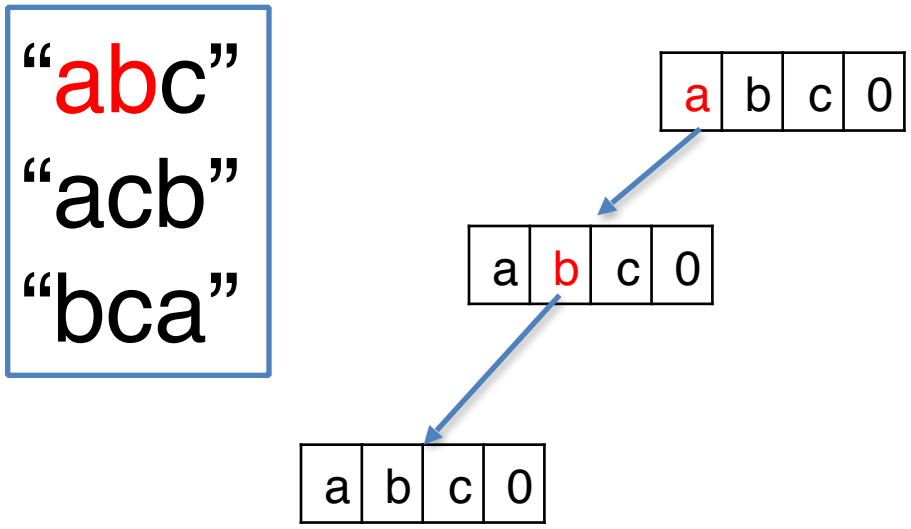
"abc"
"acb"
"bca"



made by tommy5198

Trie

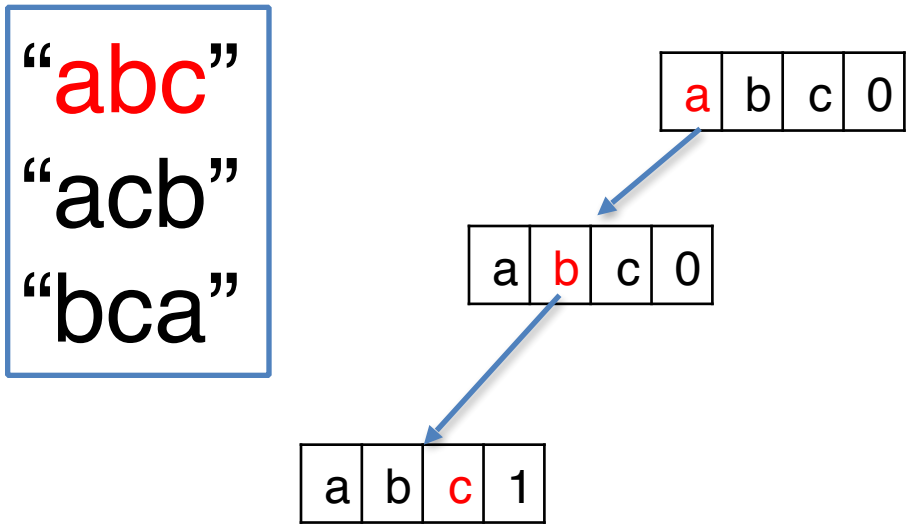
- Add string into trie



made by tommy5198

Trie

- Add string into trie

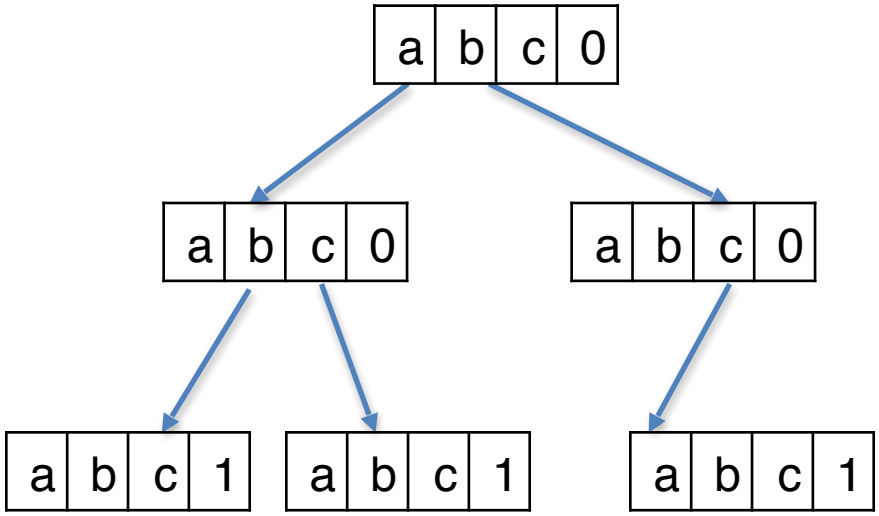


made by tommy5198

Trie

- Add string into trie

"abc"
"acb"
"bca"



made by tommy5198

Trie

- Find string in trie
  - Check counter

```
bool find(char *str) {
    Trie *tmp = root;
    while(*str) {
        if(tmp->next[*str] == NULL)
            return false;
        tmp = tmp->next[*str];
    }
    return tmp->cnt;
}
```

Learn more

- Some note
  - Duplicated string (“abc”, “abc”)
  - Null string (“”)
  - Delete all node after each testcase
  - Compress Trie
  - Static trie

Practice

POJ- 3630

made by tommy5198