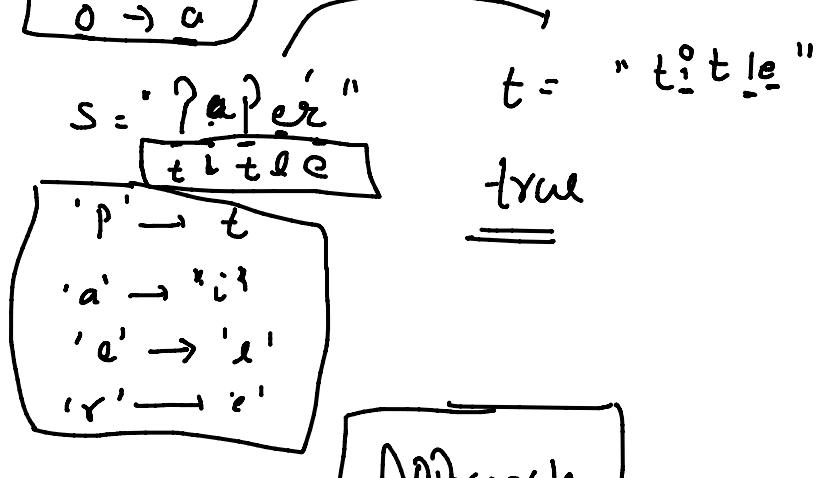
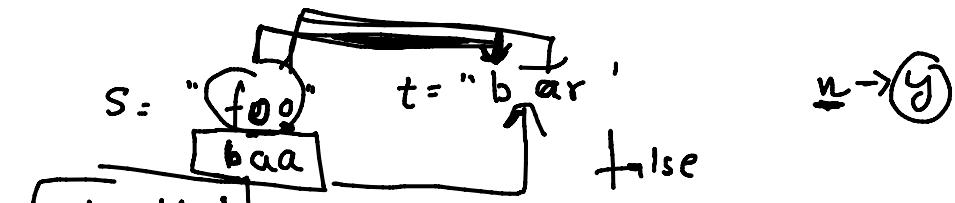
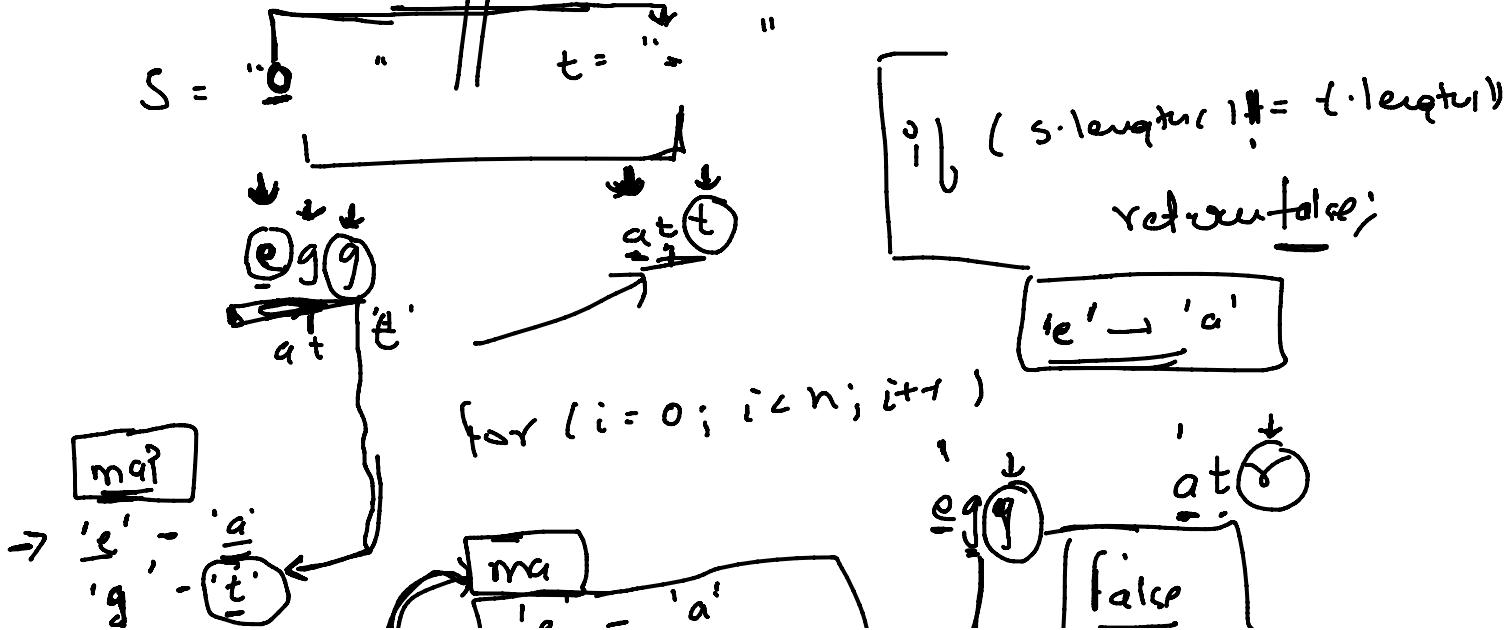
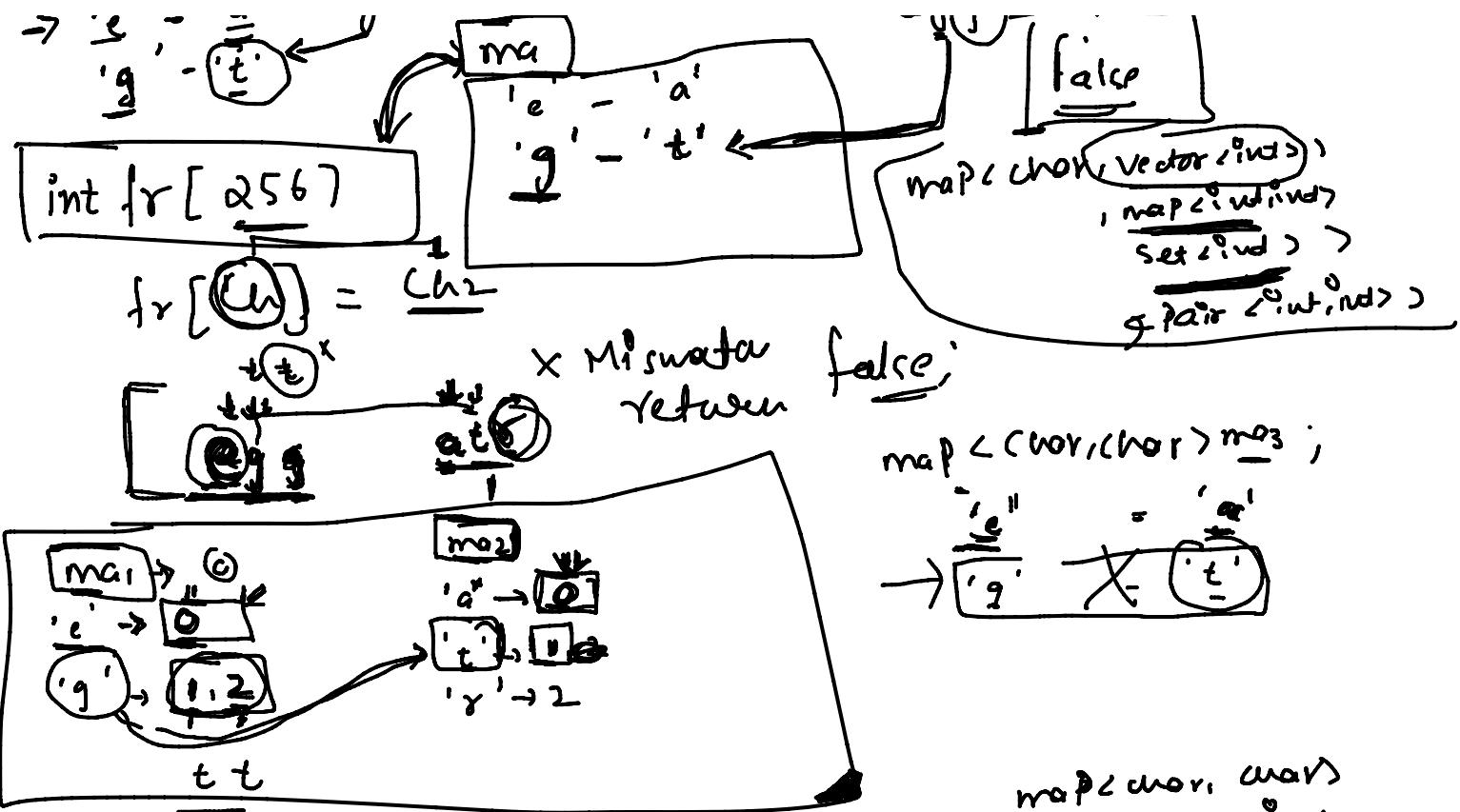


CLASS - 22Isomorphic strings

Approach





map<char, vector<int>> ma;

map<char, vector<int>> ma2;

if (s.length() != t.length())
 return false;

s[i]
forader

for (i=0; i < n; i++)
 ma1[s[i]].push-back(i);

for (i=0; i < n; i++)

ma2[t[i]].push-back(i);

map<char, char> vis;

for (i=0; i < n; i++)

if (vis.count(t[i]) != 0) and ,



ma1[char] = value

eg (t[i], i)

ma1

e - 0

ma2

a - 1

t - 1, 2

if $(\underline{\text{vis} \cdot \text{count}(f[i])} != 0)$ and
 $(\underline{\text{vis}[t[i]}] != \underline{s[i]})$
 return false;
 else if $(\underline{\text{vis} \cdot \text{count}(f[i])} = 0)$ and
 $(\underline{\text{vis}[t[i]}] == \underline{s[i]})$
 (continue);
 else if $(\underline{\text{ma}[\text{s[i]}]} != \underline{\text{ma}[\text{t[i]}]})$ // is match possible
 return false;
 $\underline{\text{vis}[t[i]}] = \underline{\text{s[i]}};$
 return true;
 ||

$\boxed{\text{ma['a']}[0]}$
 |||
 $\boxed{\text{ma['a']}[0]}$
 $\boxed{\text{ma['a']}[1]}$

egg add
 addd add
 return false

unordered-map

$\text{map} \rightarrow$ custom comparator

↳ custom map

class Person

of string name;

string last name;

Person P₁;

Person P₂;

" " P₃;

compare

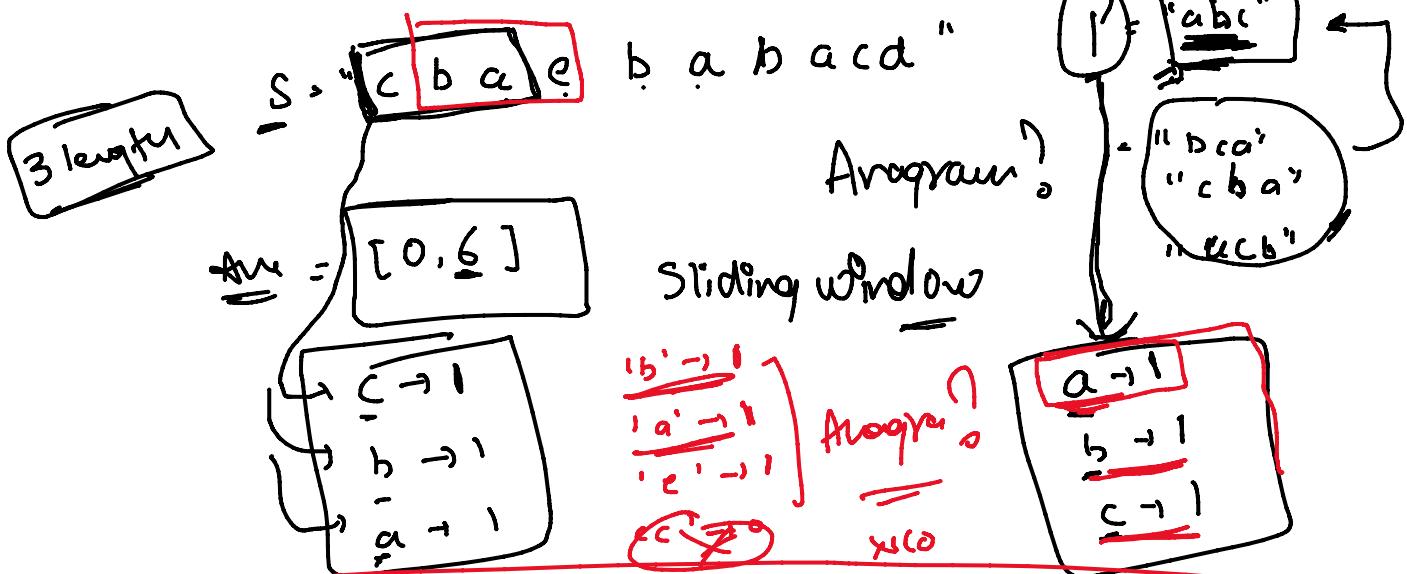
$\boxed{\text{map}} < \boxed{\text{Person}}$, $\text{int} \gg \text{ma}$

ma[P₁] = 1

Find all anagrams



Find All Anagrams



Code

```
map<char, int> ma;
for (auto i : P)
    ma[i]++;
int n = s.size();
int m = P.size();
```

map<char, int> maz;

```
For (i = 0; i < m; i++)
    maz[s[i]]++;
vector<int> arr;
```

bool flag = 1;

```
for (auto i : maz)
    if (i.second != maz[i.first])
        flag = 0; break;
```

if (flag)

(Compare
window with
original
string)

string
 if (flag)
 arr.push-back(0);
 For (i=m; i < n; i++)
 {
 ma2[s[i]] ++;
 ma2[s[i-m]] --;
 if (ma2[s[i-m]] == 0)
 ma2.erase(s[i-m]);
 flag = 1;
 for (auto j : ma)
 if (j.second == ma2[i-(int)])
 {
 flag = 0; break; } // map same
 if (flag)
 arr.push-back(i-m+1);
 window starting index

arr[0] =
 return arr;
 abcabc
 i -> a
 j -> b
 k -> c

ma2
 b -> 1
 a -> 1
 c -> 1

abc
 ma
 a -> 1
 b -> 1
 c -> 1

$a \rightarrow 1$
 $e \rightarrow 1$

$$s = \frac{\text{time to Practice}}{\text{time to Practice}}$$

$$\frac{\text{time to Practice}}{\text{time to Practice}}$$

P = ~~toc~~

P = ~~toc~~
 $(e \cdot o \cdot b \cdot c)$

variable size window

window trace
map

't' → 0
~~'i'~~

~~'m'~~
~~'n'~~

'o' → *

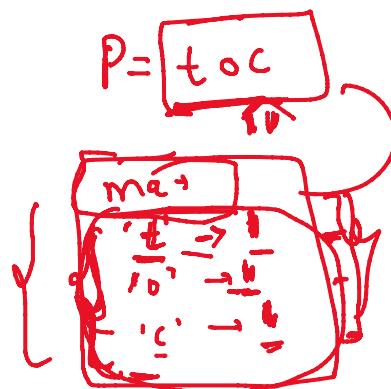
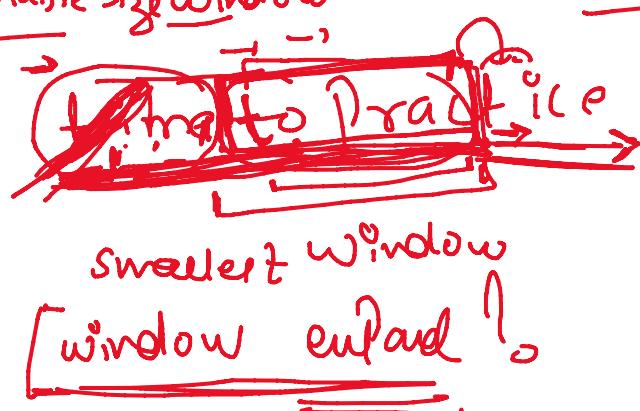
'p' → *

'r' → *

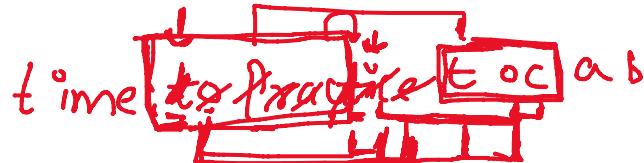
'a' → *

'c' → *

'b' → *



expansion stop?
check can we shrink?



expand
shrink?

Code

ori → s pat → t

Pat → m
s → n

map<char,int> ori;

map<char,int> Pat;

For (i=0 ; i < m ; i++)

for ($i = 0; i < m; i++$)
 $\downarrow \text{Pat}[t[i]]++;$] "reversed Map"

4

original Map
int count = 0; window kitne elements scanned

int start = -1;

int curstart = 0;

int m = n;

original Map

(S) 5

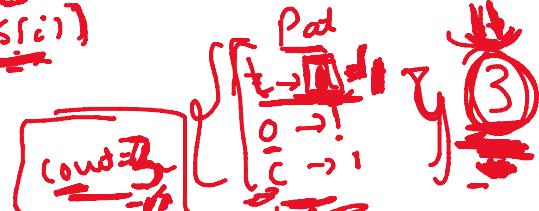
for ($i = 0; i < n; i++$) "original window

"window expand

$\downarrow \text{if } (\text{ori}[S[i]) < \text{Pat}[S[i])$

- count++;

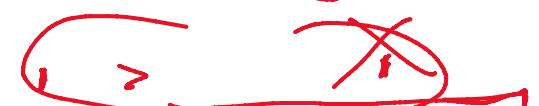
[on if] $S[i])++;$



"if [count == m)

"if shrink"

$\downarrow \text{while } (\text{ori}[S[\text{curstart}]) > \text{Pat}[S[\text{curstart}])$ or $\text{Pat}[S[\text{curstart}]) = 0$



$\downarrow \text{if shrink}; \text{ori}[S[\text{curstart}]) =;$

$\text{curstart}++;$

4

int len = $i - \text{curstart} + 1;$

$\downarrow \text{if } (m > \text{len})$

min length

$\downarrow \text{if } \text{max} < \text{len}$

$\text{max} = \text{len}$

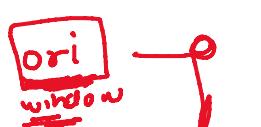
$\text{start} = \text{curstart}$

4

Home to practice

toc

$i = 0$



curstart

length

start

~7 4