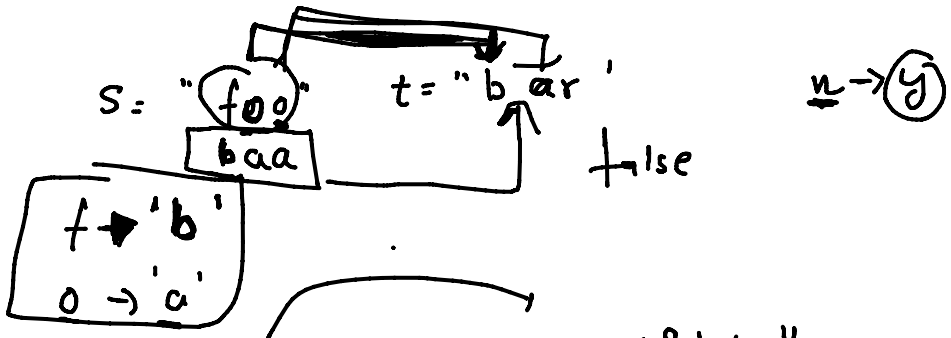
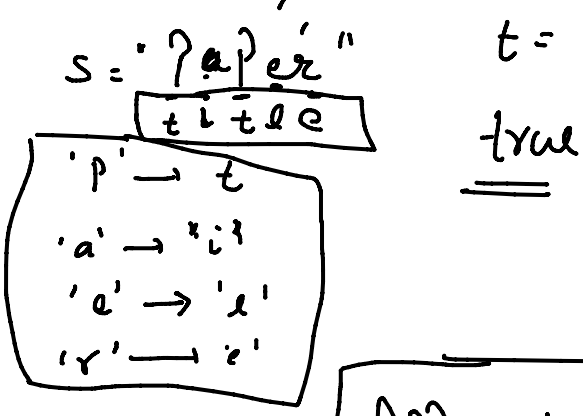
Class 22

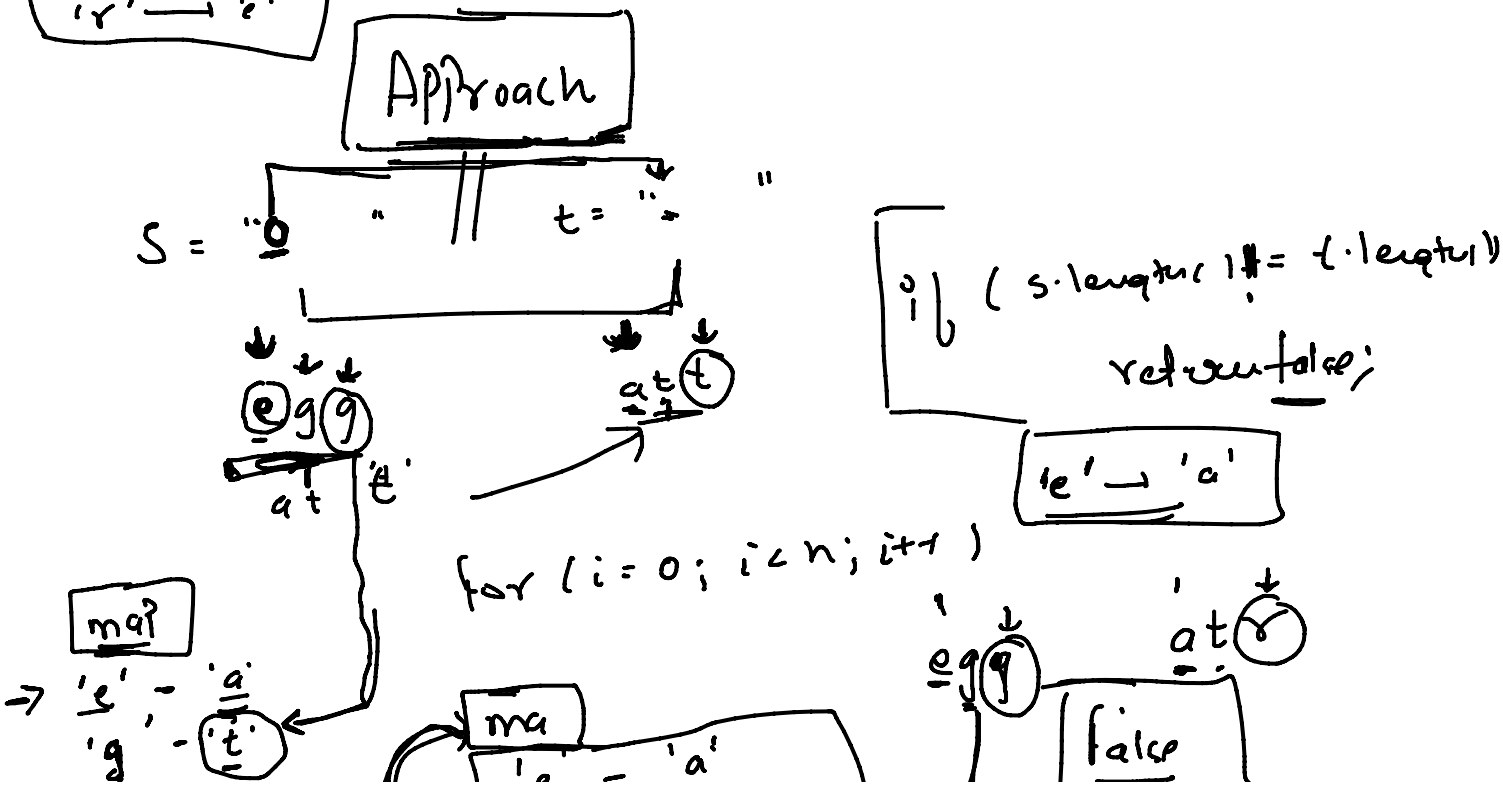
01 March 2023 20:58

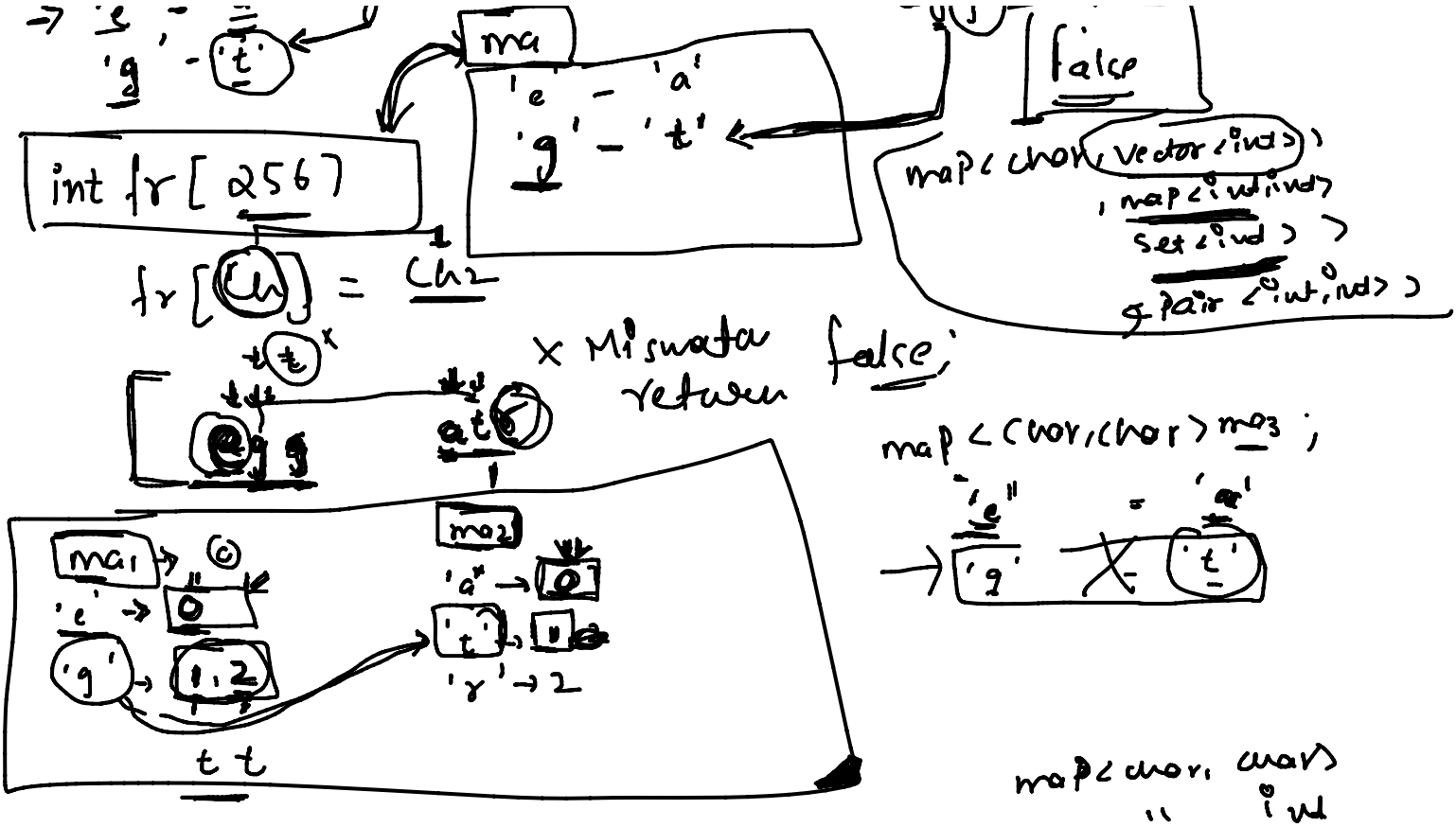
CLASS-22

Isomorphic Strings









map<char, char>
int, int

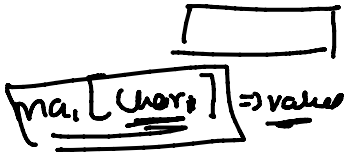
map<char, vector<int>> ma1;

map<char, vector<int>> ma2;

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

s[i]
↓
character

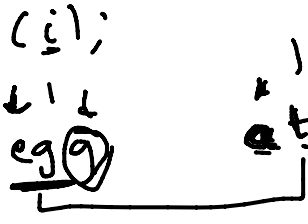
for(i=0; i<n; i++)
ma1[s[i]].push_back(i);



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
→ e → 0
g → 1,2

ma2
a → 0
t → 1,2

{ if (vis.count(t[i]) != 0 and
vis[t[i]] != s[i])
return false;

Not matching

else if (vis.count(t[i]) != 0 and
vis[t[i]] == s[i])
continue;

else if ((ma1[s[i]] != ma2 [t[i]]) // Is match possible
return false;

vis[t[i]] = s[i];
}

return true;

→ e → 0
g → (1,2)
t → (1,2)

vis
'a' - 'e'
't' - 'g'

ma['a'] [0]
ma['a'] [0]
ma['a'] [1]

egg	add
addd	add

return false

map → custom comparator

unordered-map
↳ custom Hash

class Person
{ string Fname;
string LastName;
}

Person P1;
Person P2;
; P3;

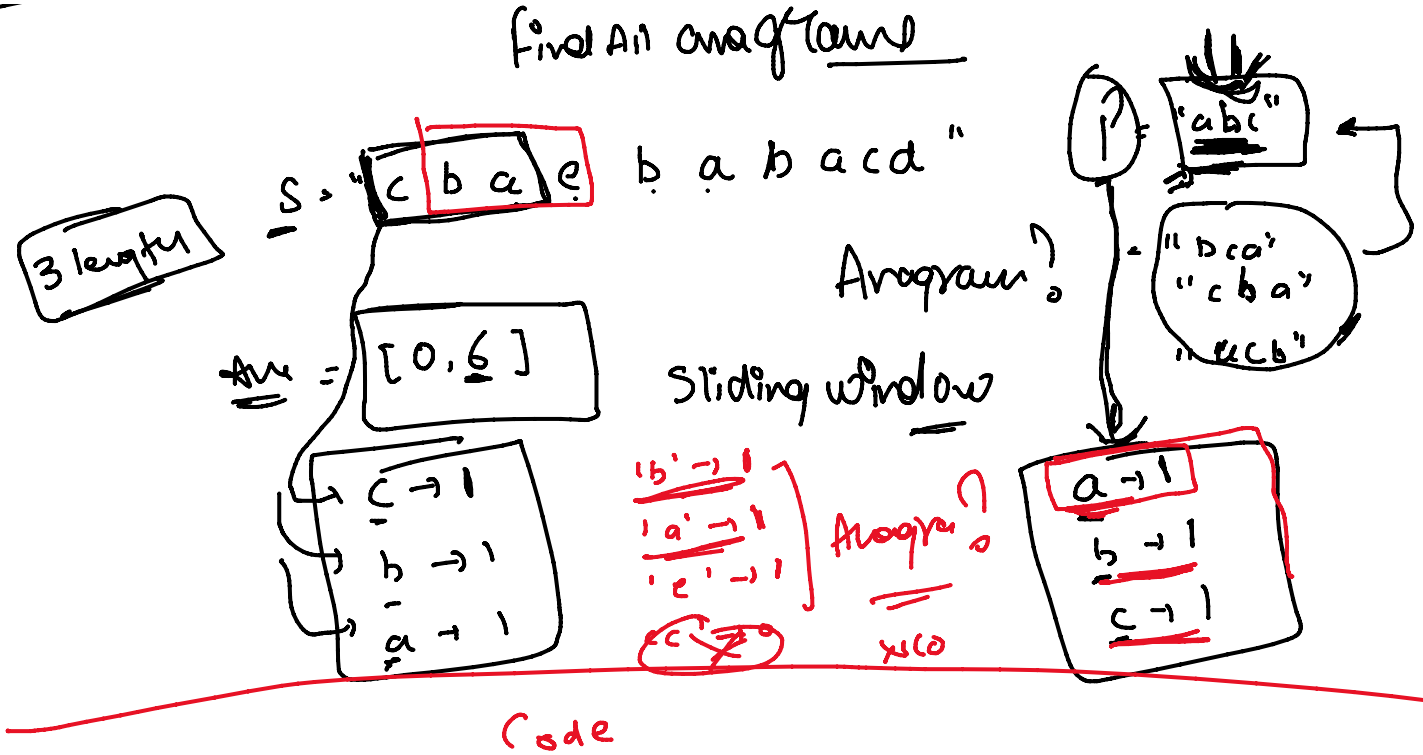
comparator

map< Person, int> ma

ma[P1] = 1

Find All anagrams

Find All anagrams



Code

```
map<char, int> ma;
for(auto i : p)
    ma[i]++;
map<char, int> ma2;
for(i = 0; i < m; i++)
{ ma2[s[i]]++;
}
vector<int> ans;
bool flag = 1;
for(auto i : ma)
{ if(i.second != ma2[i.first])
    { flag = 0; break; }
}
if(flag)
```

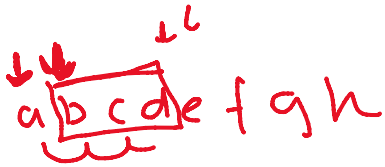
int n = s.size()
int m = p.size()

compare window with original string

String

```
if(flag)
    ans.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])
}
```



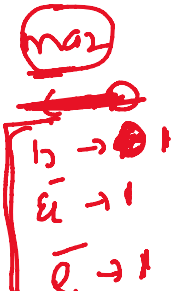
compare window with original string

```
flag = 1;
for(auto y: ma)
    if(y.second != ma2[y.first])
    {   flag=0; break; }   // map sameX
}
if(flag)
    ans.push_back(i-m+1);
```

window starting index

```
return ans;
```

ans[0]:



a → 1
e → 1

---

s = time to practice

toprac

time to practice

toprac

p = toc

p = toc

('t' 'o' 'c')

Variable Size window

window track

map

't' → 1 0

~~'i' → 1~~

~~'m' → 1~~

~~'e' → 1~~

'o' → 1

'p' → 1

'r' → 1

'a' → 1

'c' → 1

'i' → 1

smallest window

window expand ?

p = toc

use expansion stop

check can we shrink ?

a b

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++)
{ Pat[ t[i] ]++; }   // required Map
}
                                   orignal Map
int count = 0;  window kitne elements required
int start = -1;
int curstart = 0;
int mx = n;
for ( i=0; i<n; i++ )    // orignal window
                         // window expand
{
  if ( ori[s[i]] < Pat[s[i]] )
      count++;
  ori[s[i]]++;
  if( count == m)
  {  // shrink
  while ( ori[ s[ curstart ] ] > Pat[ s[ curstart ] ]
        or  Pat[ s[ curstart ] ] == 0 )
  {  // shrink;    ori[ s[curstart] ]--;
                   curstart++;
  }
  int len = i - curstart + 1;
  if( mx >= len )          min length
  {
    mx = len
    start = curstart
  }
```