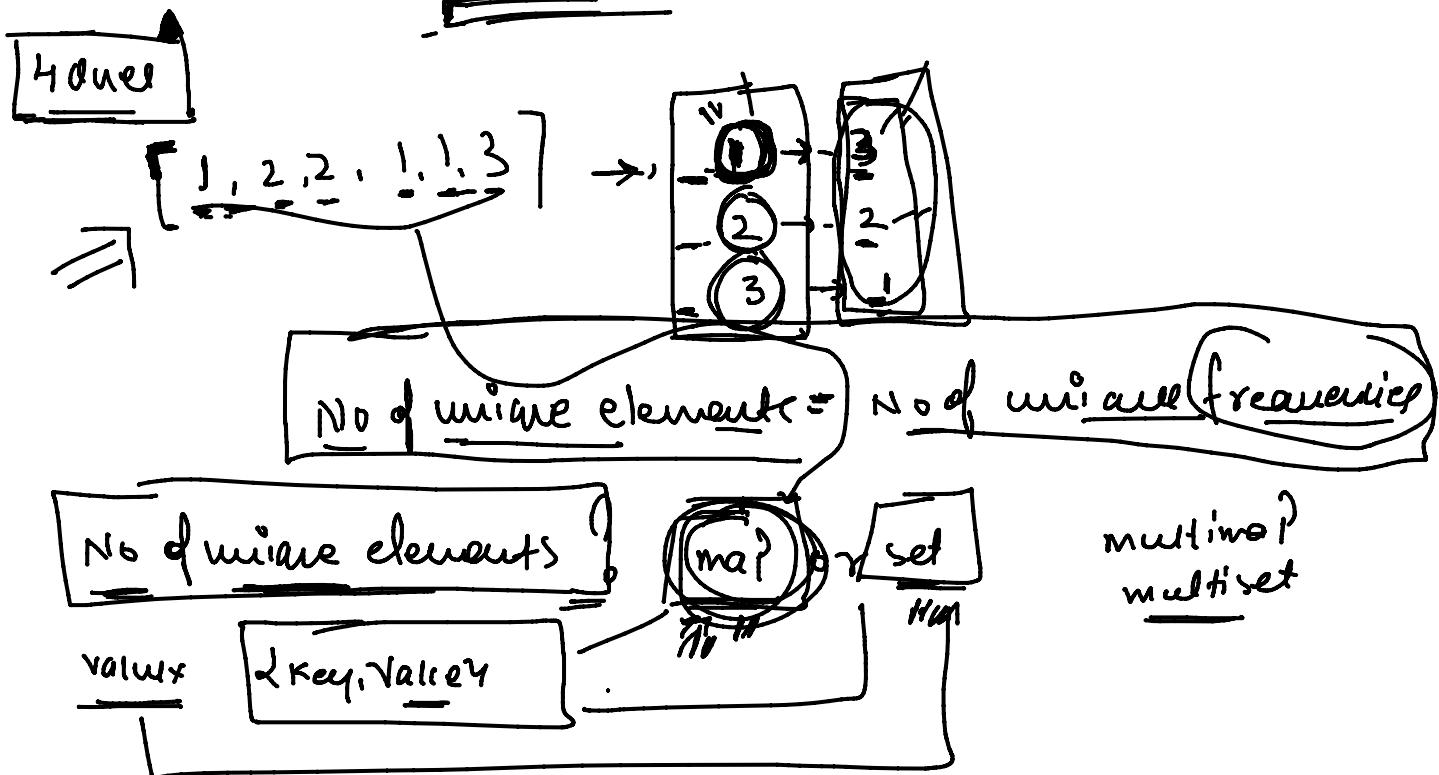


CLASS - 21

unordered_map <int, int> ma;

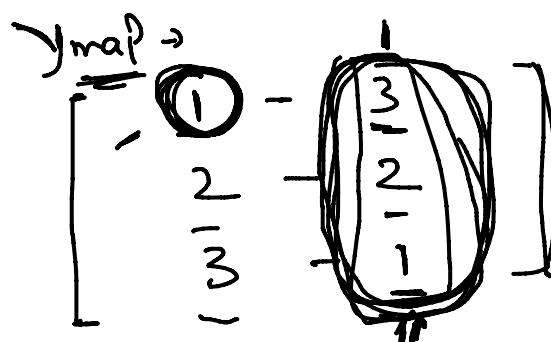
for (auto i : num) Range based loop

$ma[i]++$

~~num[i]~~

ma.size() = No of unique element

map or set



unordered_map <int, int> ma;

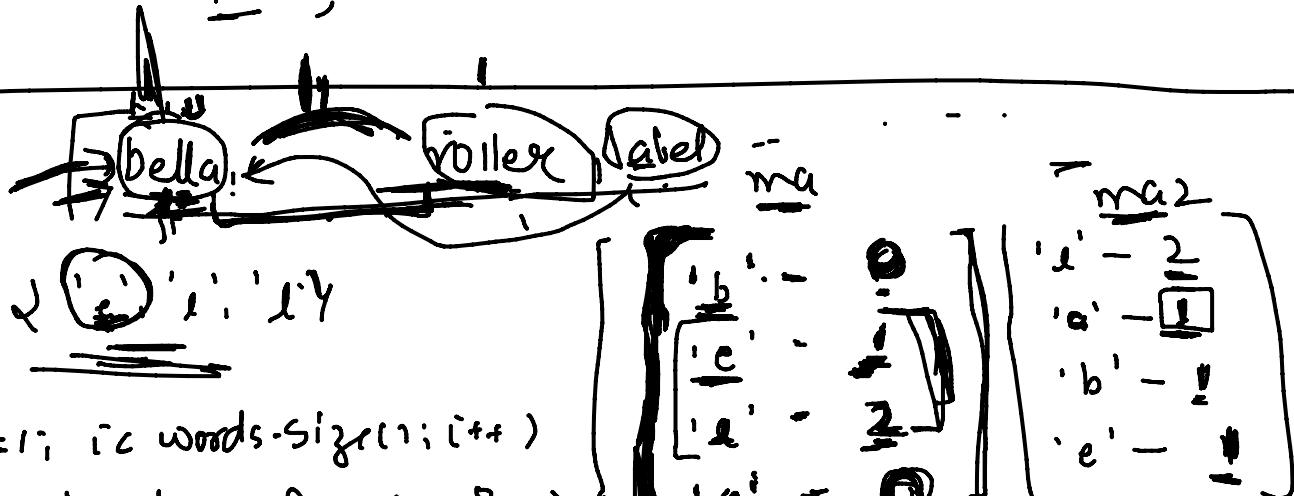
for (auto i : ma)

{ if i.first, i.second

$ma2[i.second]++$

... in a way

$\text{maz}[i.\text{second}] \leftarrow$
 If $\text{maz}[\text{size}] = \text{maz}[\text{size}]$, No of unique elements in array
 return true;
 return false;



For ($i=1$; $i < \text{words.size}(); i++$)

{ ordered-map \leq character }

for (auto j : $\text{words}[i]$) {
 $\text{maz}[j] \leftarrow$

for (auto i : maz) { Global map of string \Rightarrow }

$j.\text{second} = \min(j.\text{second}, \text{maz}[i.\text{first}]);$

y

// ma contains unique characters of all the strings

vector<string> ans;

for (auto & j : ma)

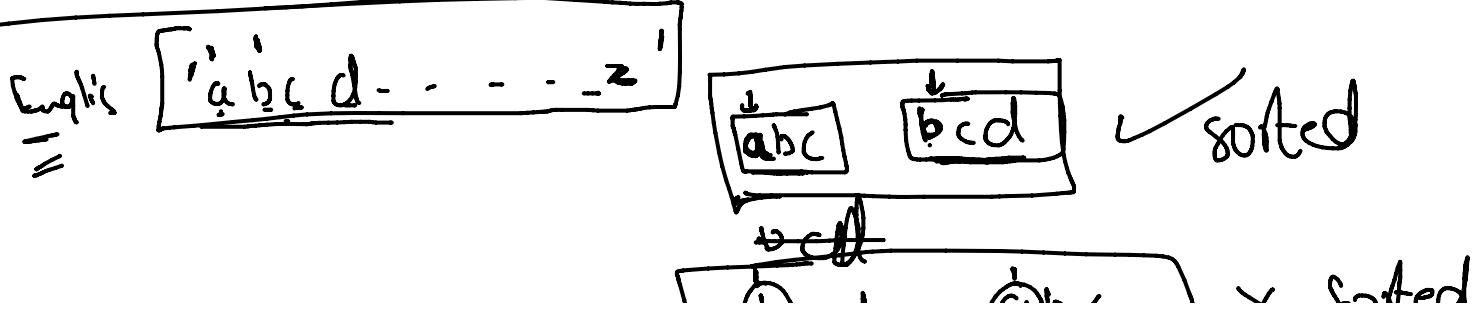
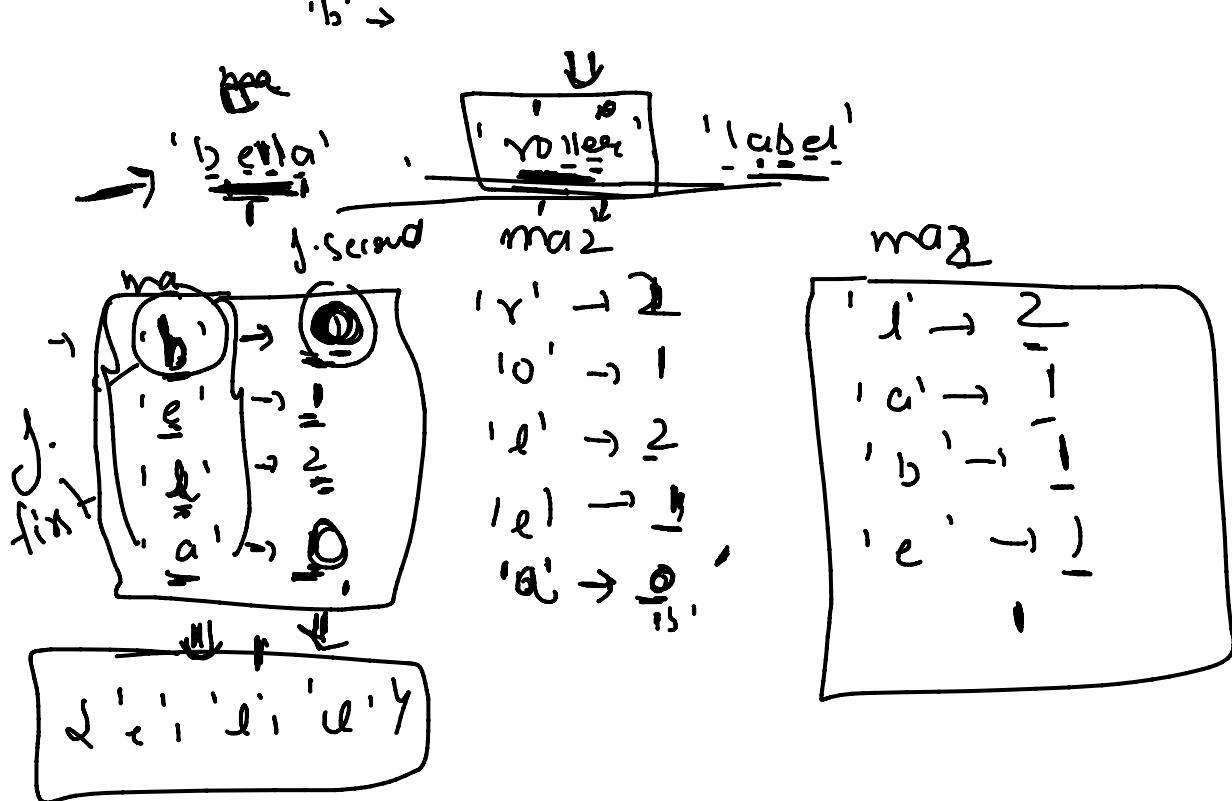
{ while ($j.\text{second}$)

```

    {
        while (j < second)
    {
        string str = " ";
        str.push_back (j . first);
        arr . push_back (str);
        j . second --;
    }
}

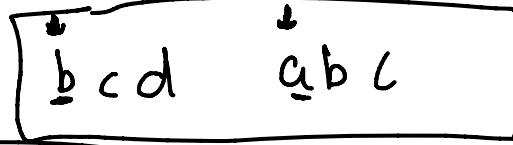
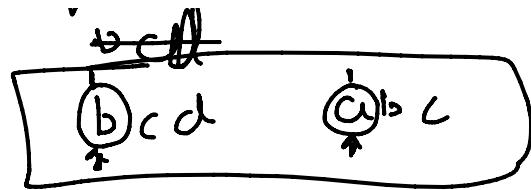
```

4
Return arr;



Alien's language:

"bacd --- zy"



to Alien
language

acb bcd cdl dbc

wow wowl

order = " " " "

map<char, int> ma;

character → freq

→ = " b a d c f g e --- yz "

'b' - 0

'a' - 1

'd' - 2

'c' - 3

.

}

bcd abc

B
sorted

unordered_map<char, int> ma;

For (i=0; i < order.size(); i++)

{ ma[order[i]] = i;

(1) (2) (3) (4)

bcd abc

cdb abc

For ($i=1$; $i < \text{words.size}()$; $i++$)
 { int $mn = \min(\underline{\text{words}[i].size()}, \underline{\text{words}[i-1].size()})$
 bool flag = 1; } sorted

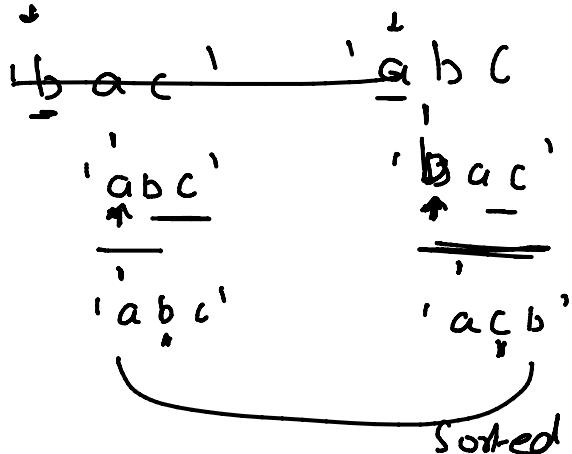
$i, i-1$
 for ($j=0$; $j < mn$; $j++$)
 { if ($\underline{\text{ma}[\text{words}[i-1][j]]} \leq \underline{\text{ma}[\text{words}[i][j]]}$)
 { break; } $j \leq mn$
 else if ($\text{ma}[\text{words}[i-1][j]] > \text{ma}[\text{words}[i][j]]$)
 { flag = 0; break; } $j \leq mn$
 if (flag == 0) return false; } $i = 3 \leq mn$

else {
 if ($j = mn$)
 { if ($mn = \underline{\text{words}[i-1].size()}$)
 continue;
 else return false; }
 if ($(i-1).length > mn$)
 { $(i) = mn$
 sorted }

else continue; } } main loop → everything in
 alignment($i, i-1$) sorted

$y \leftarrow \text{common} \sqcup$ mainboj \rightarrow Prepending is aligned
align $i:i-1$ sorted

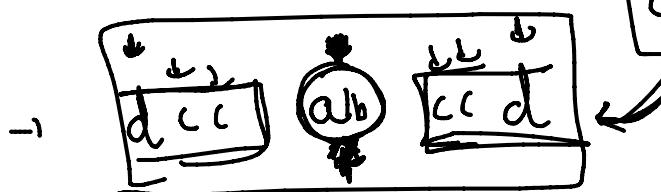
y



$i+1, i$

first unequal character
(compare them)

"ab c c c d d" \Rightarrow



character $0:0, 2:2, 0:0$

$$\begin{aligned} \text{ans} &= 0 \\ \text{curr} &= 2 + 4 + 1 \\ &= 7 \end{aligned}$$

$d \rightarrow 2 \Rightarrow$
 $a = 4$
 $b = 1$
 $c = 1$
 $e = 1$

longest subarray with sum = 0
No of " " " " = 0

Plj - k

1, 2, 3, 2, 2, -3, 1, 7, 5, 6, 8

Duffixes ... common \sqcup Duffixes $i:j$ = Prefixes $i:j$

P1

6

$$\underline{\text{PrefixSum}} = \boxed{\text{store in map}}$$

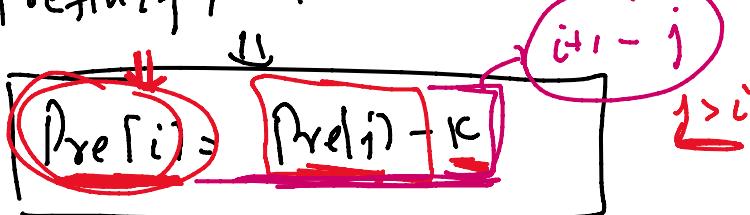
↑

$$\text{PrefixSum}[j] = \underline{\text{PrefixSum}[i]}$$

$\hookrightarrow (i+1 - 1) = \underline{\text{Sum}}$

No of Subarrays with sum \underline{K}

$$\text{Prefix}[j] - \text{Prefix}[i] = K$$



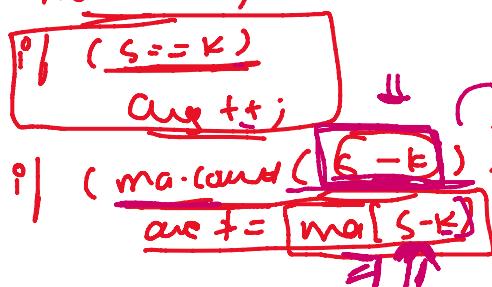
$$\text{Prefix}[i] = \text{Prefix}[i]$$

map<int, int> ma;

$$\text{int } s = 0;$$

$$\text{int cur} = 0;$$

for ($i=0$; $i < n$; $i++$)
& store \underline{s} in $\underline{\text{ma}}[i]$;



$\underline{\text{ma}}[\underline{s}] \underline{++};$

γ

return cur;



resSum = 2

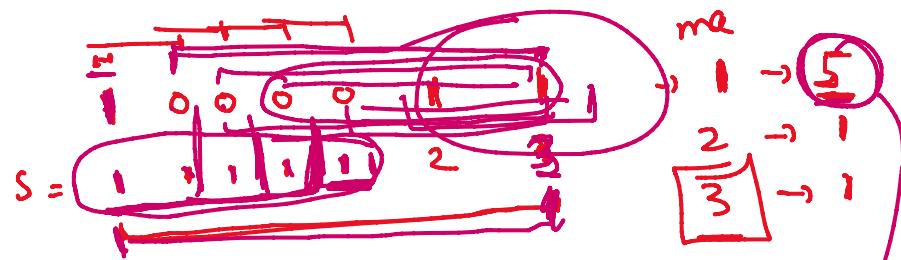
ans = 1

Sub = 2

$K = 2$

$$\underline{\text{ma}}(3 - 2) = 1$$

$$\begin{array}{r} 1 \\ \rightarrow 1 - 1 \\ 2 - 1 \end{array}$$



`ma.count(3 - 2) or ma.count[1]`

$$p[6] - p[0] = 2$$

$$p[6] - p[1] = 2$$

$$p[6] - p[2] = 2$$

$$p[6] - p[3] = 2$$

$$p[6] - p[4] = 2$$