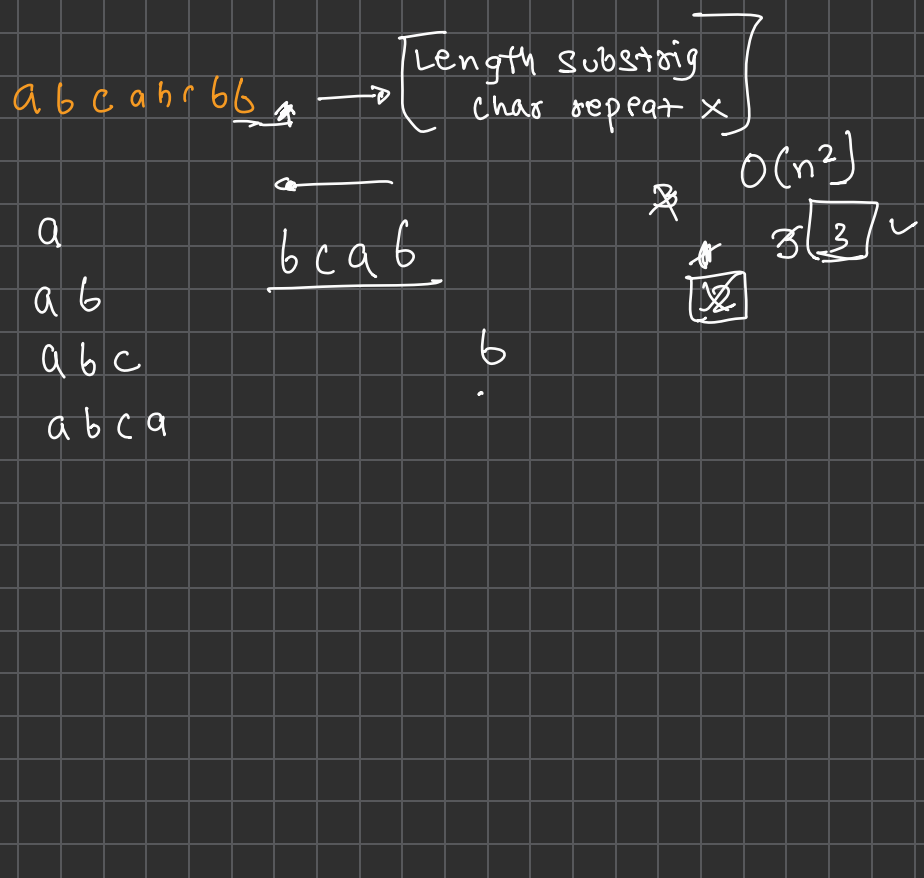
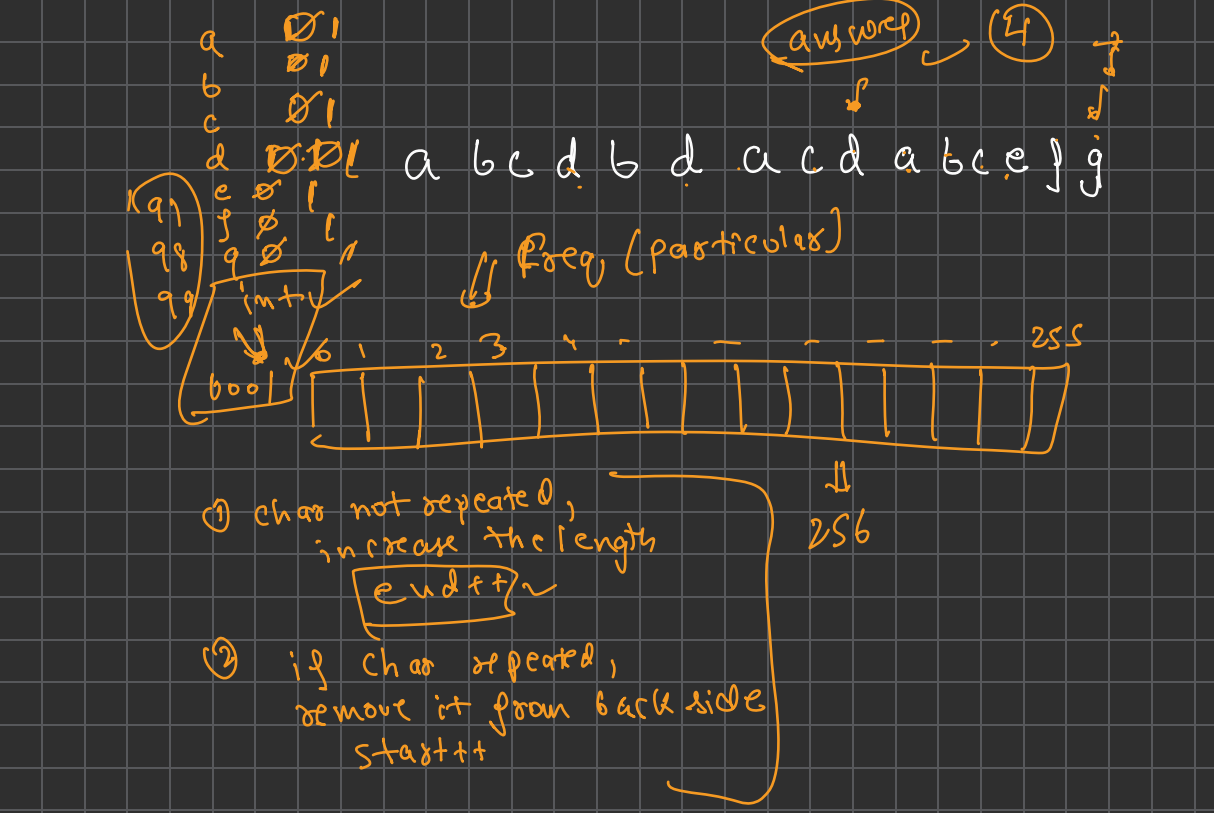
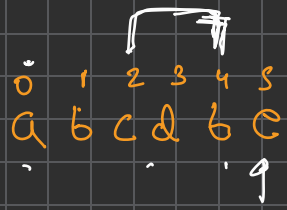

abcabcbb
Length substring
char repeat x
O(n²)
bcab
a
ab
abc
abca
b


a
b
c
d
e
j
q
a b c d b d a c d a b c e j j
answer
4
Freq (particular)
int i
bool
0 1 2 3 4 255
256
① char not repeated, increase the length
end++
② if char repeated, remove it from back side
start++



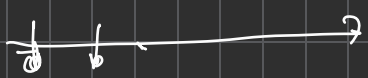
cdb

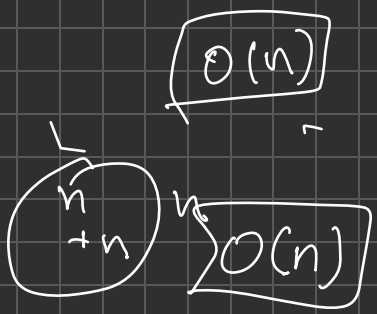
remove the last chart from our window

```
while (end < n)
{
    while (freq[s[end]])
    {
        freq[s[start]] = 0
        start ++
    }
    freq[s[end]] = 1
    end ++;
    answer = max(answer, end - start)
}
```

a = 0
b = 0
c = 1
d = 1
e = 0

answer = 1 2 3 4





unique = 0 1 2 3 4 ⇒ Total unique chars

a - 0 1 a a b c b c d b c a
b - 1 2 3 2 1
c - 1 2 3 2 1
d - 0 1 0

Smallest window length such that saare chars present ho at least once

bcdbca

Total unique = 1 2 3 4

mil jaayegi

Unique char = 4

a a b b c d

2 3 4 5

start = 0
end = 0
answer = 765

a a b b c d d

a = 2
b = 2
c = 1
d = 1

total unique char = 4

```
while (end < n)
{
    count[str[end]]++;
    if (count[str[end]] == 1)
        totalUniqueChar ++
    while (uniqueChar == totalUnique(char)
    {
        answer = min (answer, end - start + 1)
        count[str[start]]--;
        if (count[str[start]] == 0)
            totalUnique--;
        start++;
    }
    end++;
}
```

I V X L C D M

1 5 10 50 100 500 1000

-1+5 ⇒ (4)

IV = 4

XXII = 22

10+10+1+1

(22)

XLL = 70

-10+50 ⇒ 40

↓

XL = 40

CDI = 401

-100+500+1 ⇒ (401)