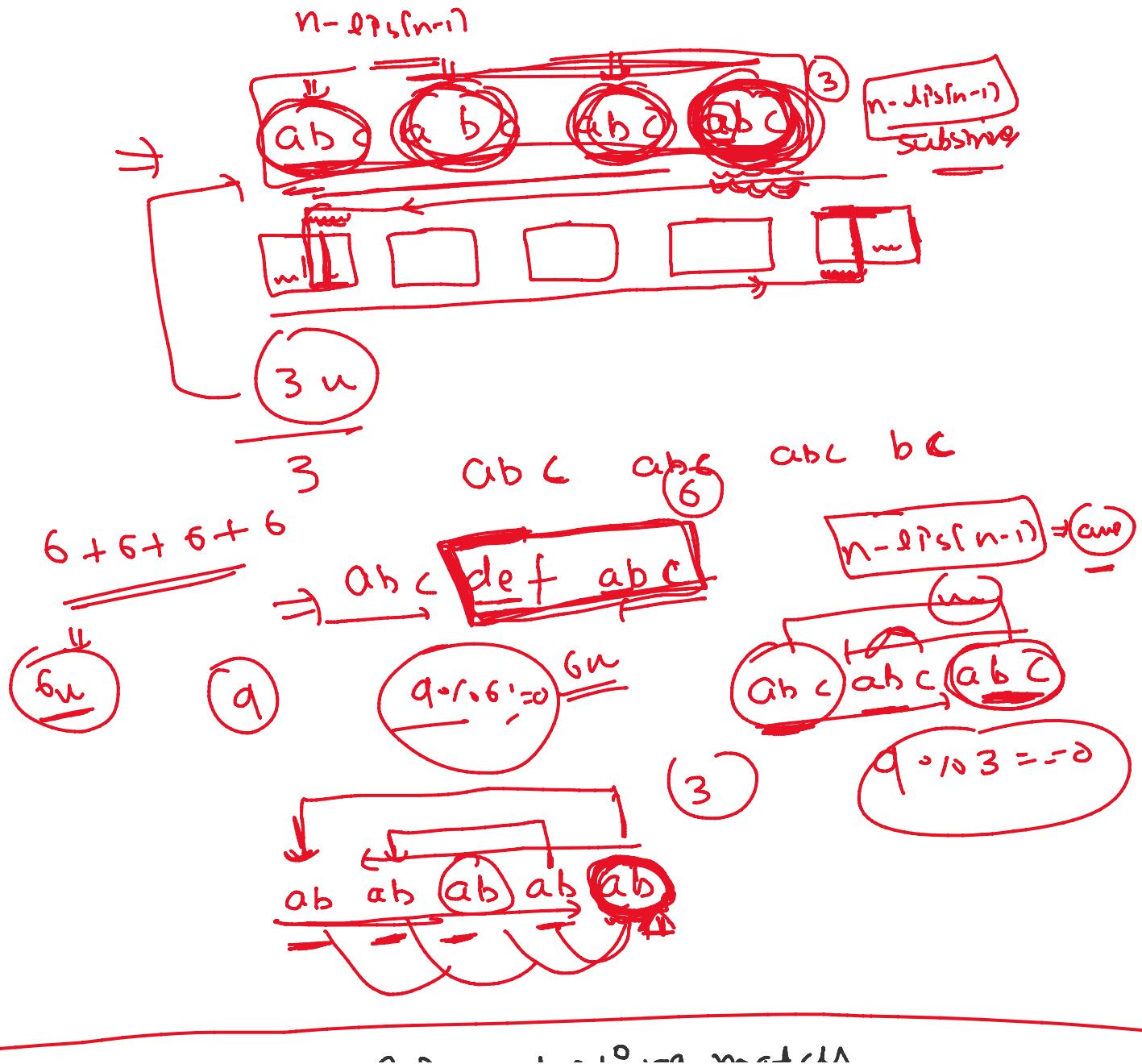
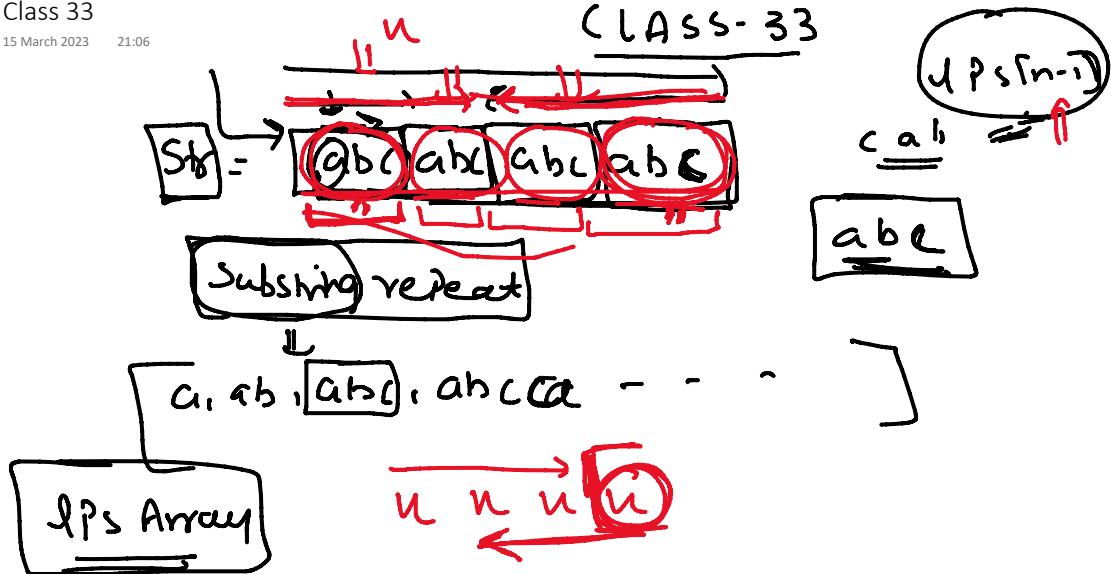
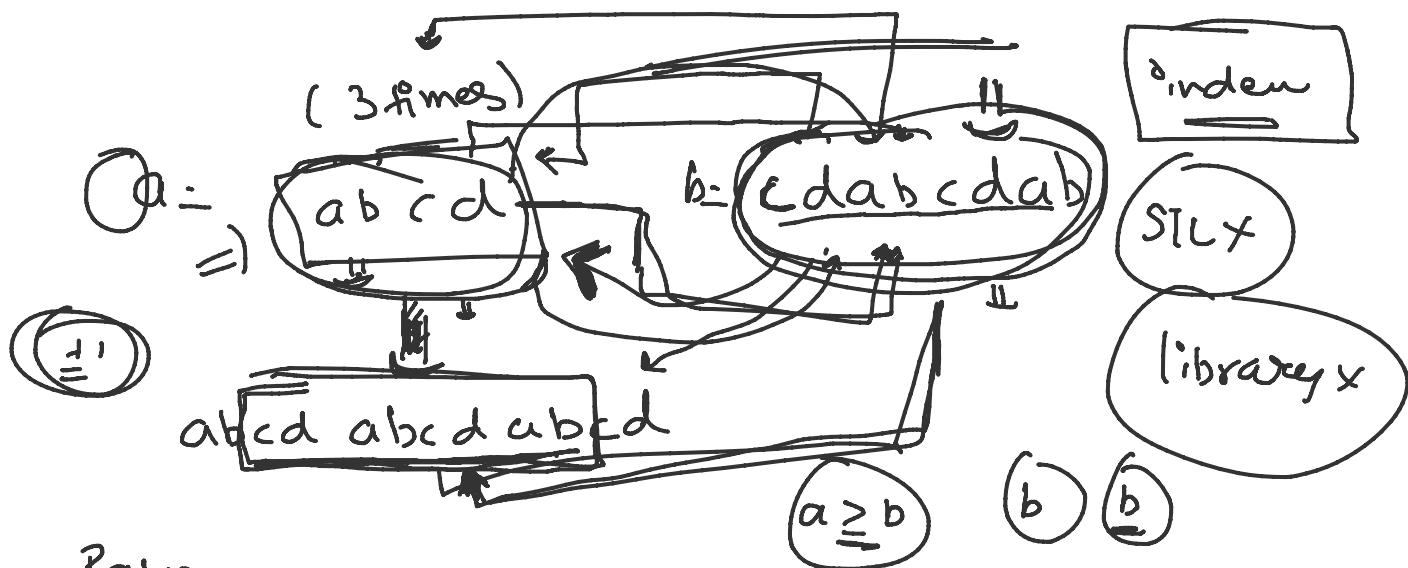


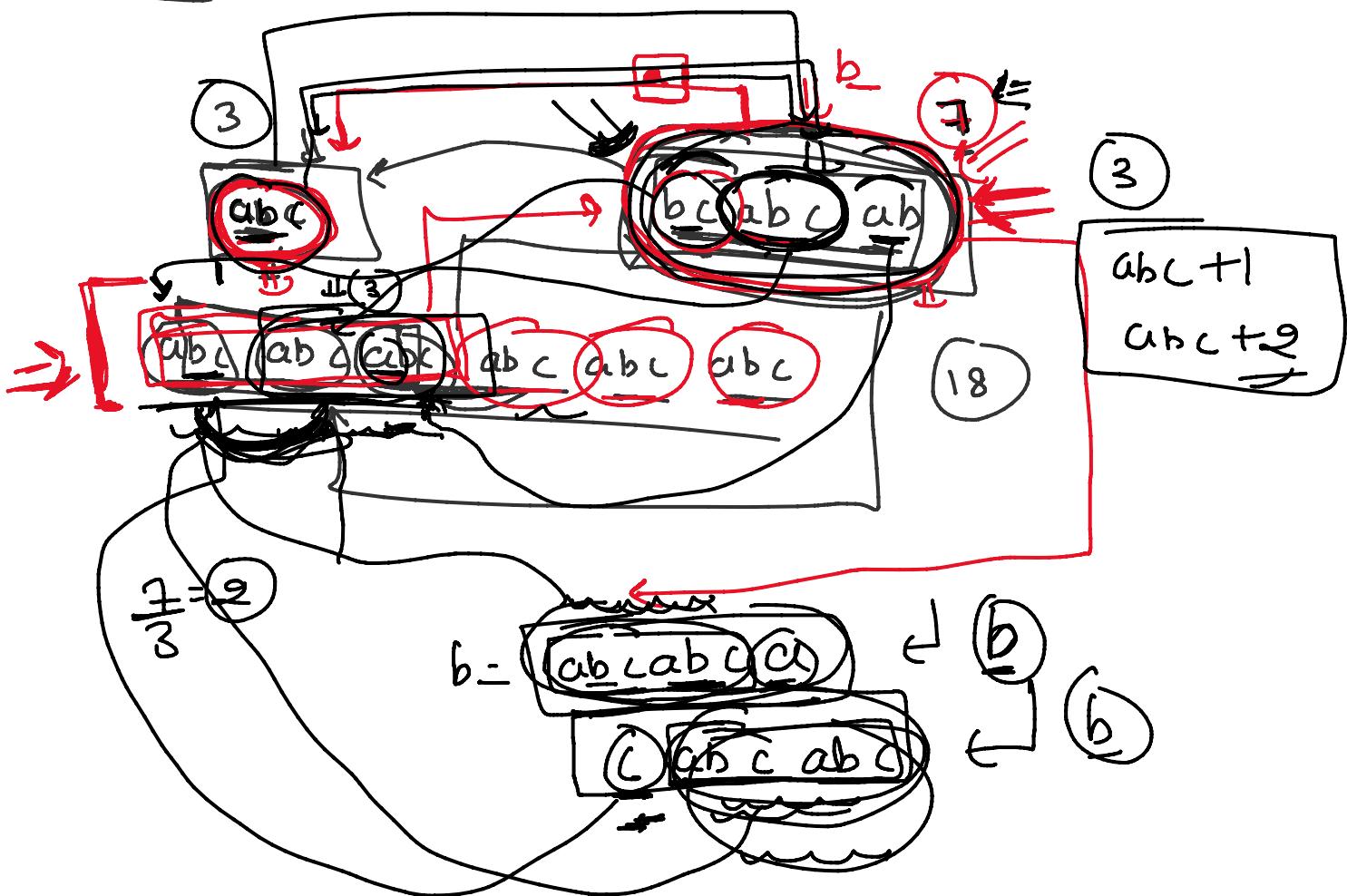
CLASS- 33



Repeated string match



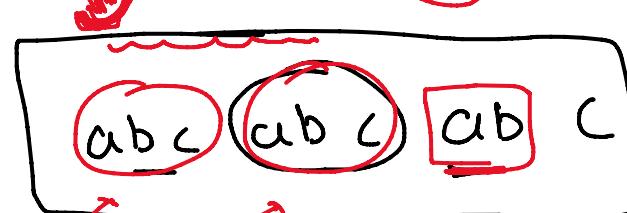
Pattern matching



\Rightarrow — $abc = n$
 — abc $= n$ —

abc $= n$

(3)



$ab \subset abc$

ab c ab c ab

b c ab c ab

abc =

$ab \subset abc$

targeted

b c ab c c

abc

abc abc ab c

(3)

$n-1$

2 approaches

abc

T.C = $O(n)$

b c ab c d a

abc ab c

1st occ \rightarrow 2nd

last occ \rightarrow $n-1$

1st index

(at)occ \rightarrow 3rd

$abc \rightarrow$ 1st occ \rightarrow 0

last occ \rightarrow $n-1$

abc ab

abc abc abc

abc ab c $n-1$

(2)

2

lattice $\rightarrow n^{-1}$

abc abc

a: abc

1:3

KMP=?

b: bcabcabcab

overlap

b c  while (i < n)
{
 cc ab**C** b; if (b[i] == a[i])

$$2 \quad \text{int temp} = 1;$$

$$\text{Round } \underline{\text{occ}} = q$$

while (b[t + emp] == a[i])

{ tempt;

1 + + i

if ($t = m$)

~~cout++ ; j = 0;~~

```

    ↳
    ↳ i = count++; j = 0;
    ↳
    ↳ i = temp;
    ↳

```

2nd

```

int m = a.size();
int n = b.size();
int ans = 1;
string temp = a;

```

```

while (a.size() < b.size())
    ↳ a += temp;
    ↳ ans++;

```

```

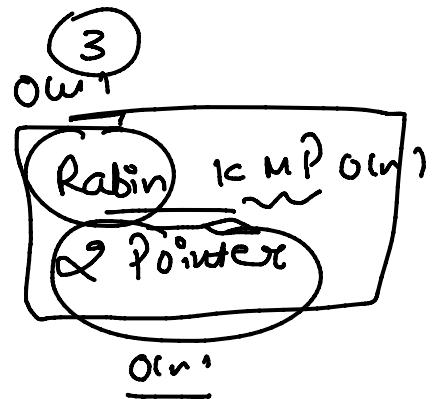
    ↳
    ↳ a ≥ b
    ↳
    ↳ abc + abc → b = abc abc
    ↳
    ↳ b = cabcab → using rmp
    ↳
    ↳ abc abc abc ←

```

```

else
    ↳ a += temp;
    ↳ if (check(a, b))
        ↳ return ans;
    ↳

```



a b a b

a b a a b a b

a a b a b a a b a b

Shortest Palindrome

ζ^T Palindrome

insert

str = "a a c e c a a a"

a b a

Catforcer

KMP

LPS

N a m a n

i = j

a a a c e c a a a

a a a c e c (a a a) ? a a a c e c a a a

a a c e c a a a a a c e c a a a

dPS \Rightarrow

a a b a a b ③

d c b a b c d ? d c b a

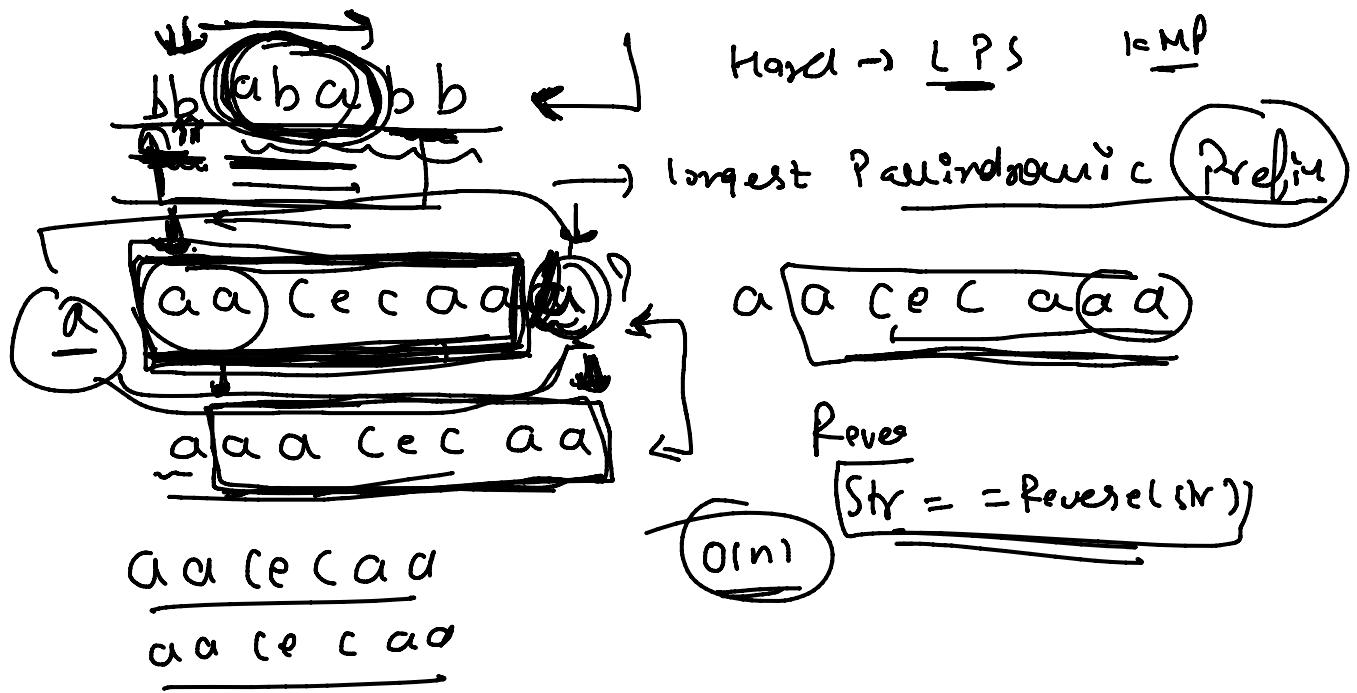
d. Palindrome \rightarrow

\rightleftharpoons

a b a

a b a

a a a c e c a a a ? a a a c e c a a a



aacecaa ? aa a c e c a d

aa a c e c a a
 ↙ ↘
 a a b ? b a a
 ↙ ↘
 2 pointer

a a b ? b a a .
 b a a
 Pallindrome

b a b c ? c b a b

↙ ↘ ↙ ↘
 d c b a b c d ? d c b a
 ↙ ↘ ↙ ↘

d c b a b c d ? d c b a b a
 ↙ ↘ ↙ ↘
 Pallindrome

(LPS)

↙ ↘ ↙ ↘
 a b a b c a ? a c b a b a

Pattern Matching

T.C = O(nm)

lcs = o(mn)

b b a b a b b b b a b a

largest Palindromic Prefix

bb a b a b b +
bb a b a

bb ab a b b