Knuth Morris Pratt (KMP)
Algorithm for String Matching

Lecturer Akib Zaman, Dept. of CSE, UIU

T: a b a b c
0 1 2 3 4

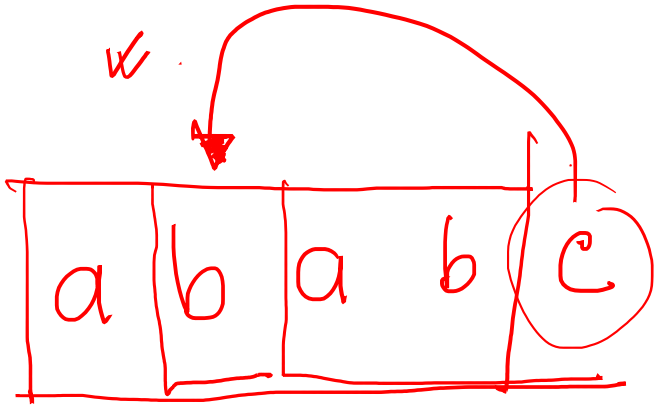
M: bc

T[i] == M[j] Match
both index forward.

→ else

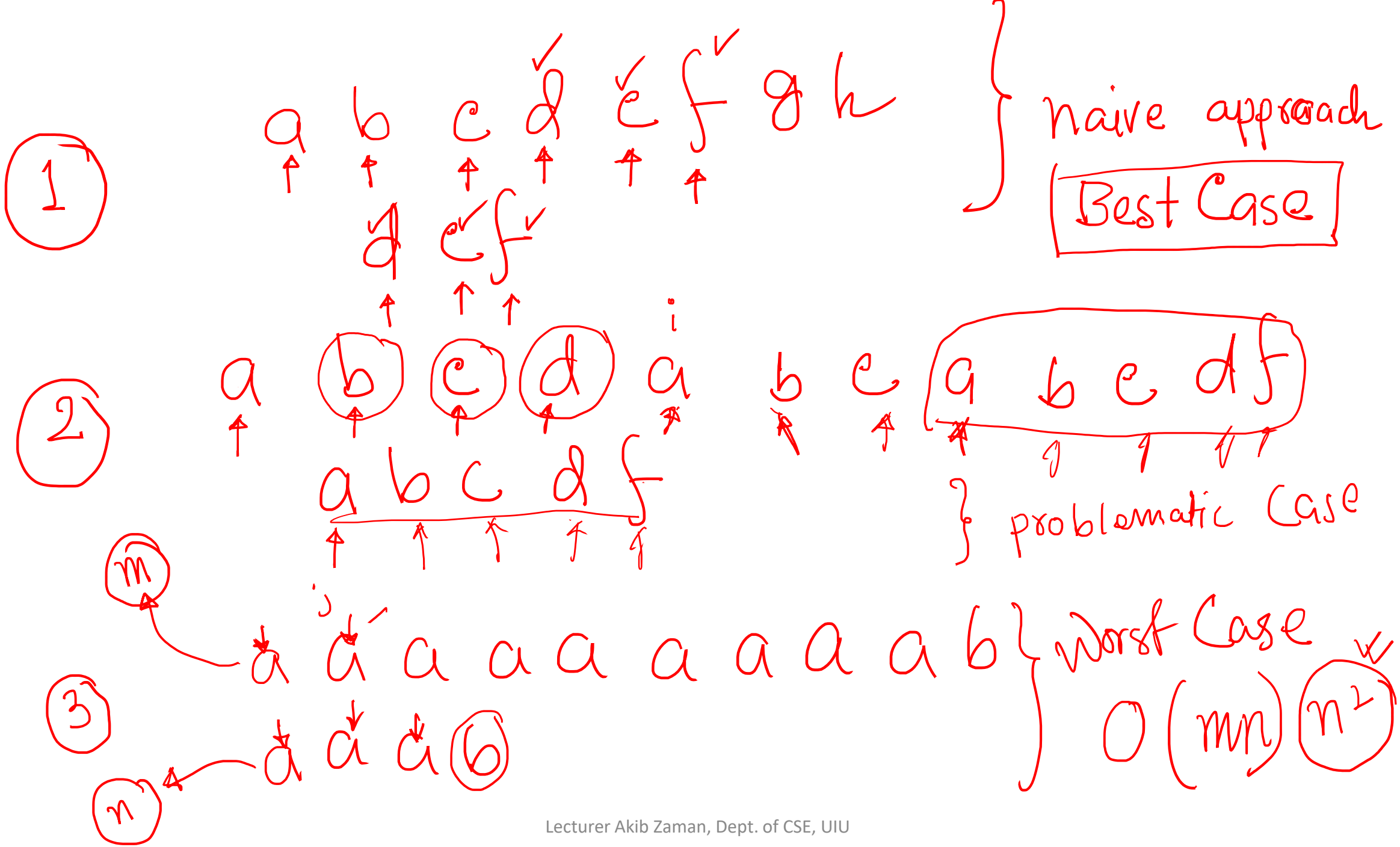
i → forward

3 → 4



a b c a b a b d

a b a b d


1
a b c d e f g h
naive approach
Best Case
2
a b c d a b c a b c d f
i
a b c d f
problomatic Case
3
m
n
a a a a a a a a a b
d d d b
Worst Case
O(mn)
n²

Suffix: Last

prefix: First

akib

i b

bi

akib

a k

ak

* table / LPS

① a b c d a b c a b f

0 0 0 0 1 2 3 1 2 0

② a b c d c a b f a b c

0 0 0 0 1 1 2 0 1 2 3

Match

len++

i++

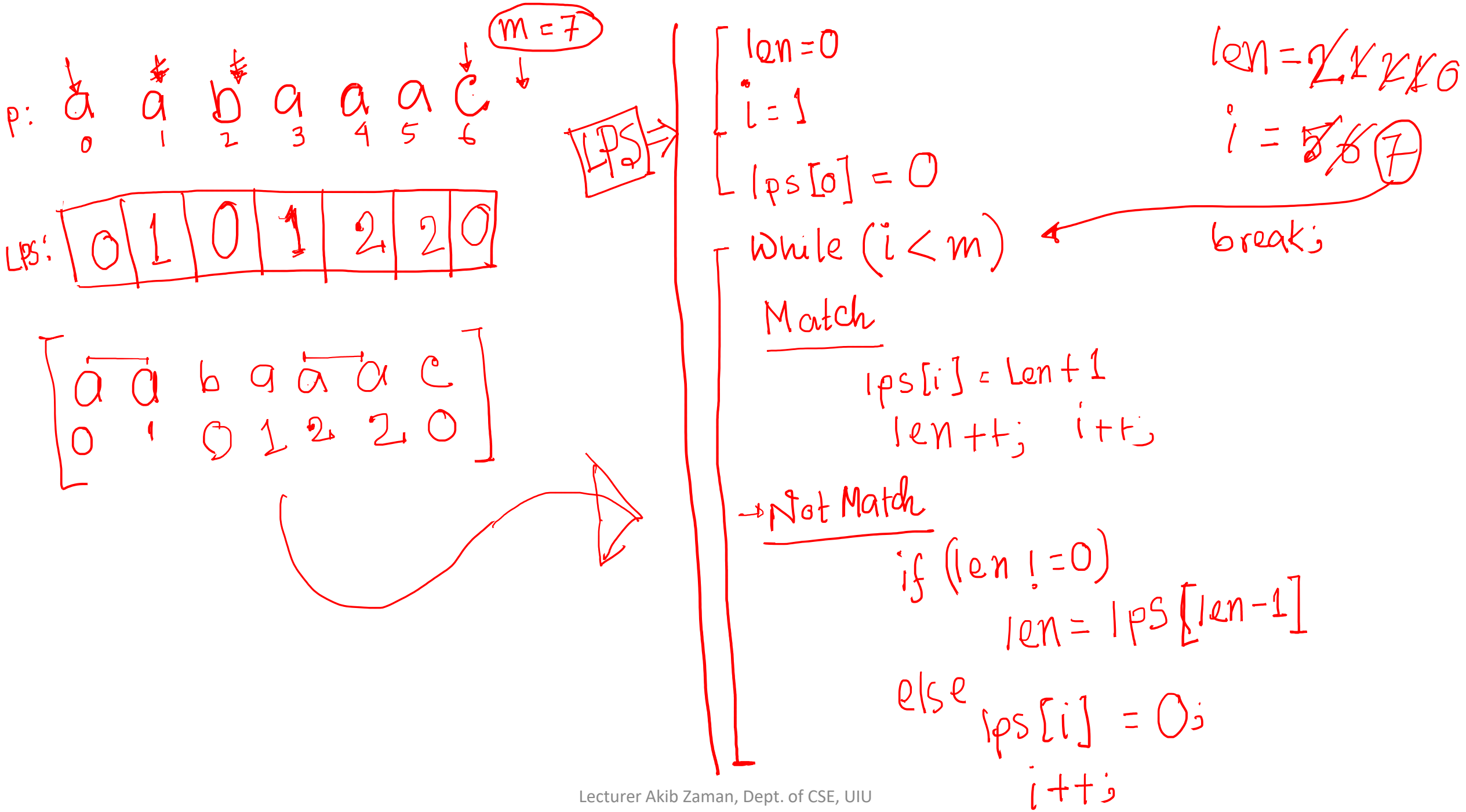
Not Match

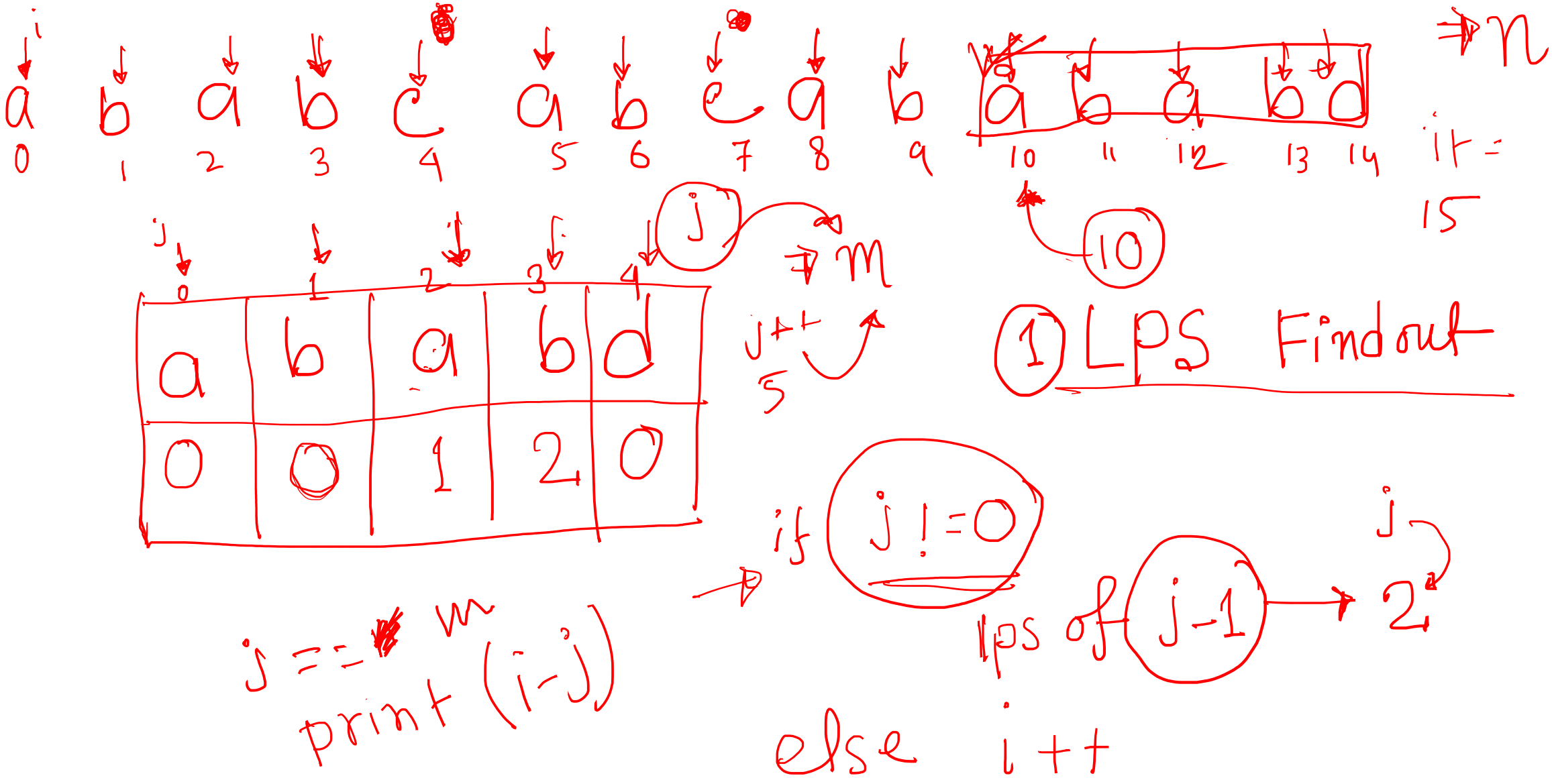
i++

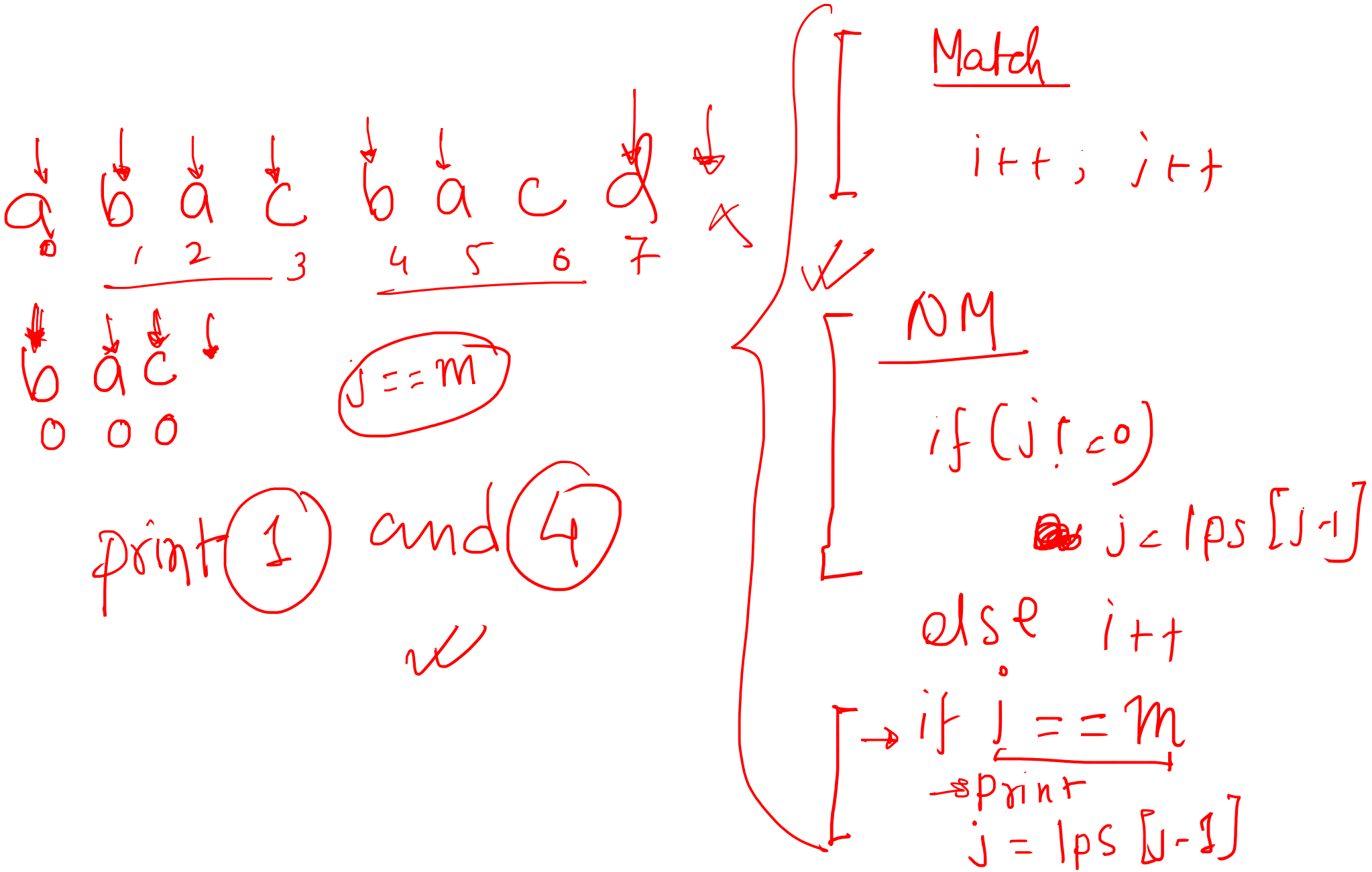
a a b a a a c

len = 0 + 1 i

0 1 0 1


m = 7
p: a a b a a a c
0 1 2 3 4 5 6
LPS:
0 1 0 1 2 2 0
a a b a a a c
0 1 0 1 2 2 0
LPS
len = 0
i = 1
lps[0] = 0
While (i < m)
len = 2 X X X 0
i = 5 6 7
break;
Match
lps[i] = len + 1
len++; i++;
Not Match
if (len != 0)
len = lps[len-1]
else
lps[i] = 0;
i++;


a b a b c a b c a b a b a b d
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
i
n
it = 15
j
m
10
j++
5
1 LPS Find out
a b a b d
0 0 1 2 0
if j != 0
lps of j-1
j
2
i++
else
j == m
print (i-j)


a b a c b a c d
0 1 2 3 4 5 6 7
b a c
0 0 0
j == m
print 1 and 4
Match
i++ ; j++
NM
if (j != 0)
j = lps [j-1]
else i++
if j == m
print
j = lps [j-1]