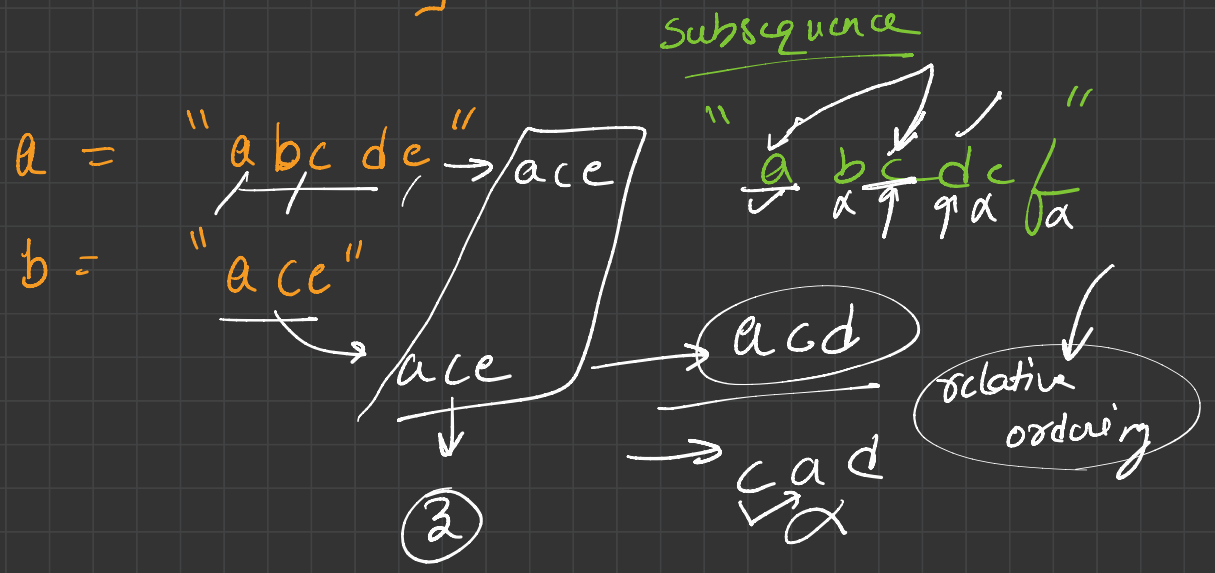
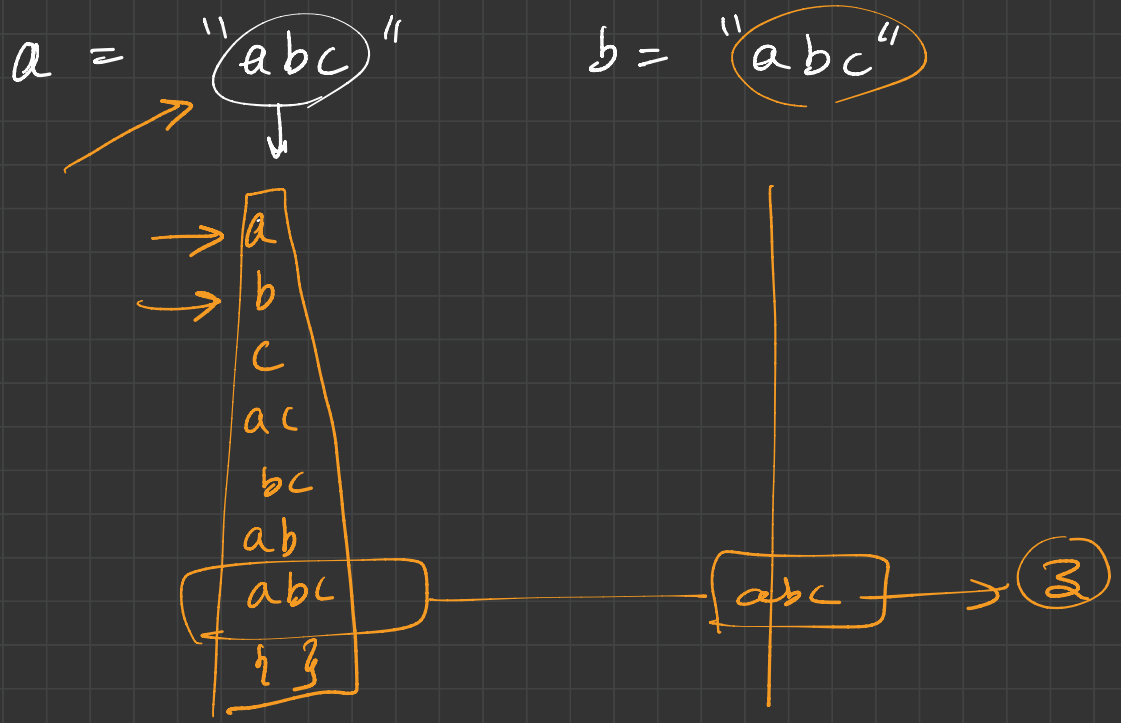
Dynamic Programming

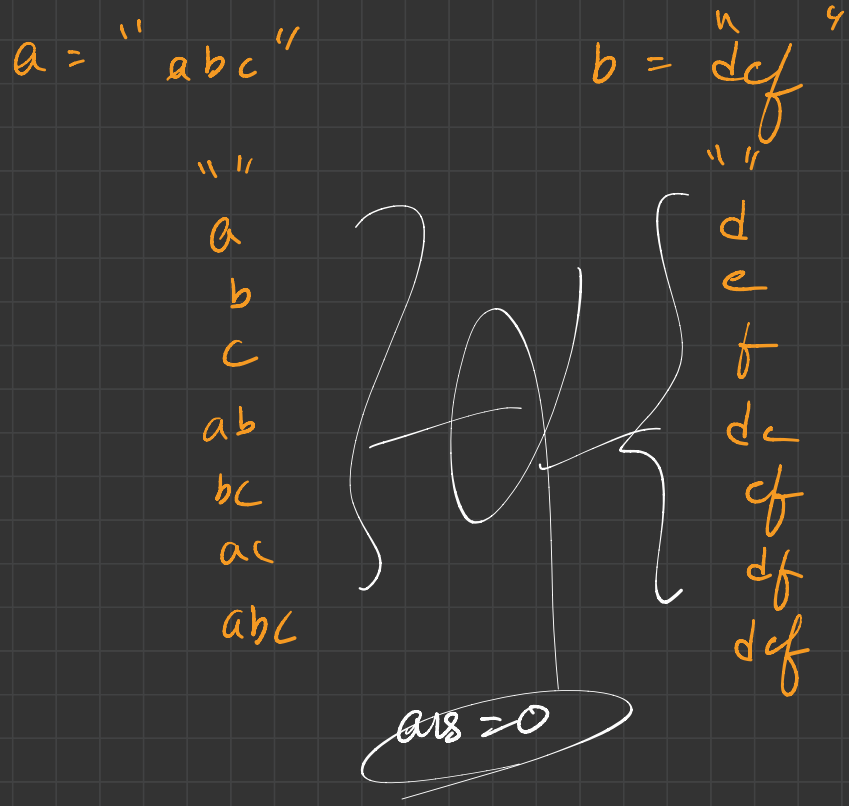
Common Subsequence

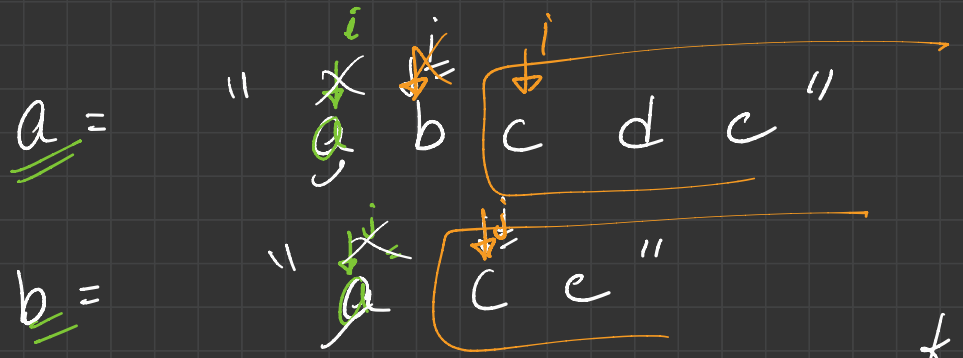
→ [Longest Common Subsequence]

Subsequence



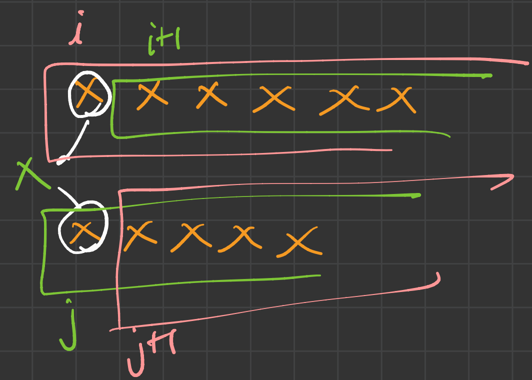

a = "abc"
b = "abc"
a
b
c
ac
bc
ab
abc
abc
3


a = "abc"
b = "def"
""
a
b
c
ab
bc
ac
abc
""
d
e
f
de
df
ef
def
ans = 0



$f(a, b, i, j)$

```
if (a[i] == b[j]) // match
    return 1 + f(a, b, i+1, j+1)
else
    return max [ f(a, b, i, j+1)
                 f(a, b, i+1, j) ];
```



Top-Down

i → 0 → n → a.len

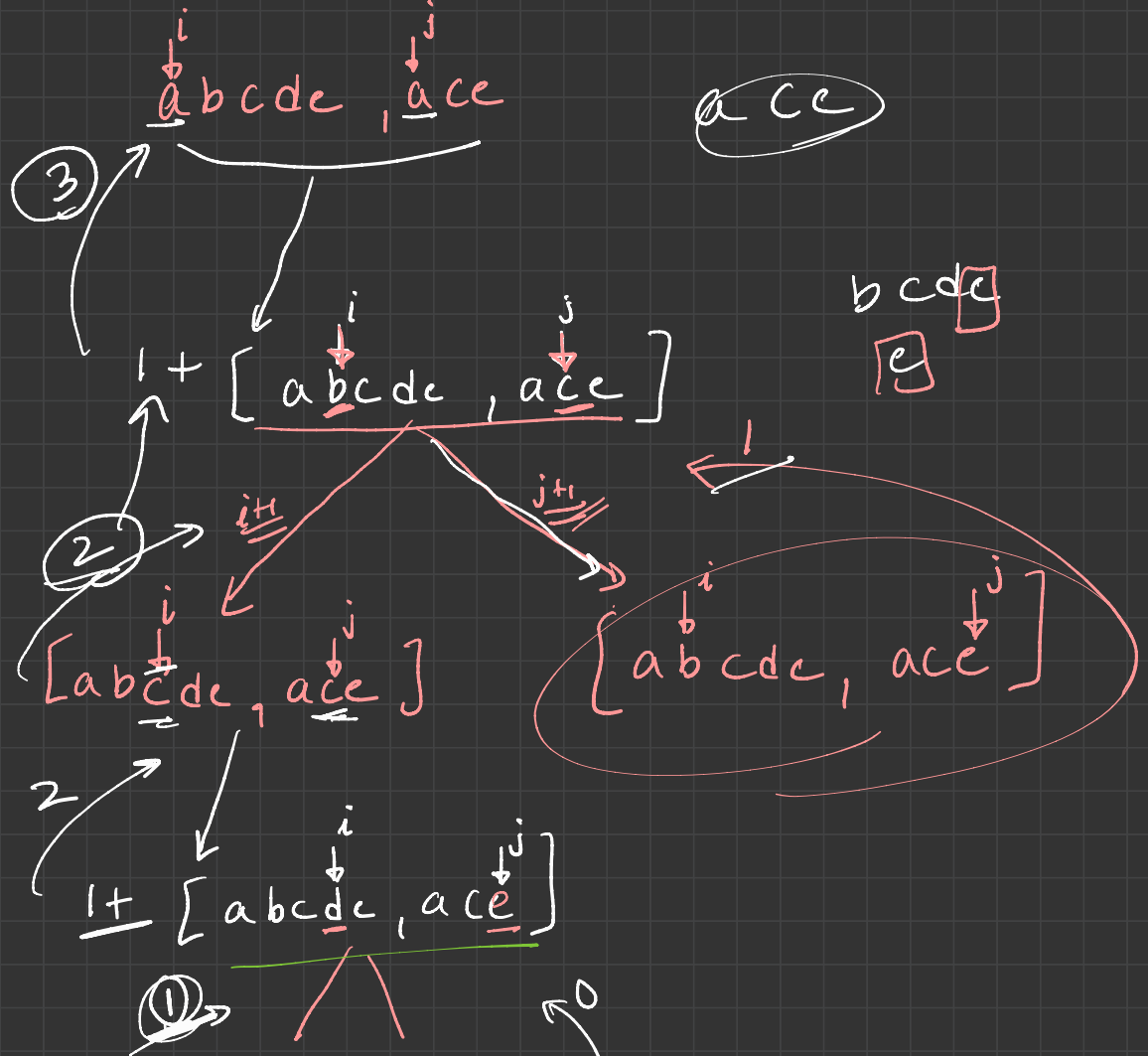
j → 0 → m → b.length

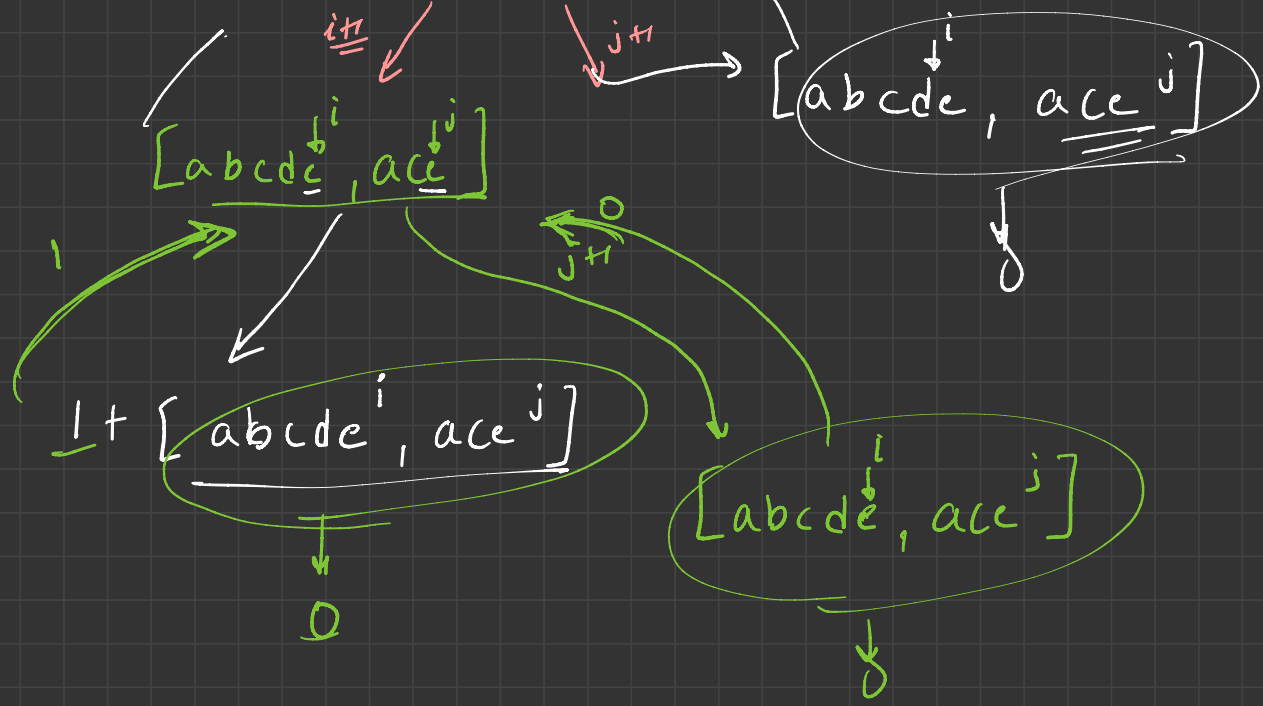
curr → next

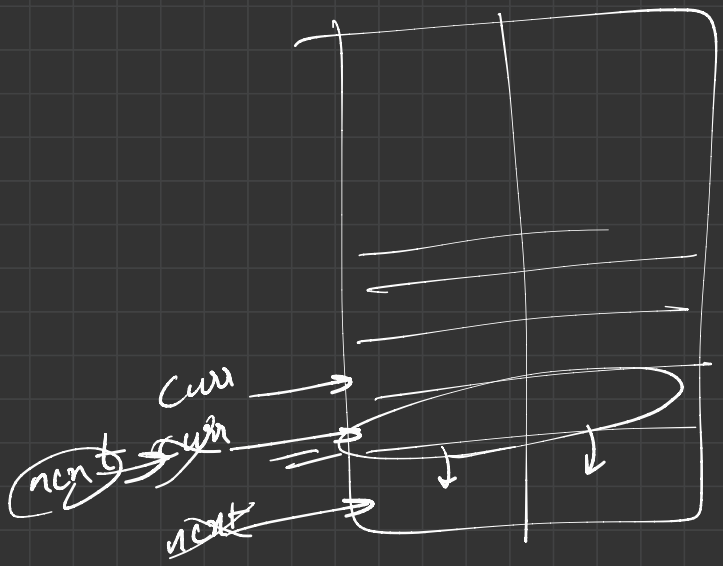
B.U

i → n → 0

j → m → 0


i
j
abcde , ace
ace
3
bcdg
e
1+ [abcde , ace]
i+1
j+1
1
2
[abcde , ace]
[abcde , ace]
2
1+ [abcde , ace]
1
0


i+1
j+1
[abcde, ace]
[abcde, ace]
1
0
j+1
1 + [abcde, ace]
0
[abcde, ace]



after each it:

next = curr