print all arrangement of characters (ie. permutation of word): → (All characters are unique)

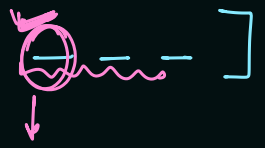
word → abc → Total no. of permutation = $n!$

= $3!$ = $3 \times 2 \times 1$ = 6 arrangements.

- abc
- acb
- bac
- bca
- cab
- cba

permutations

a, b, c



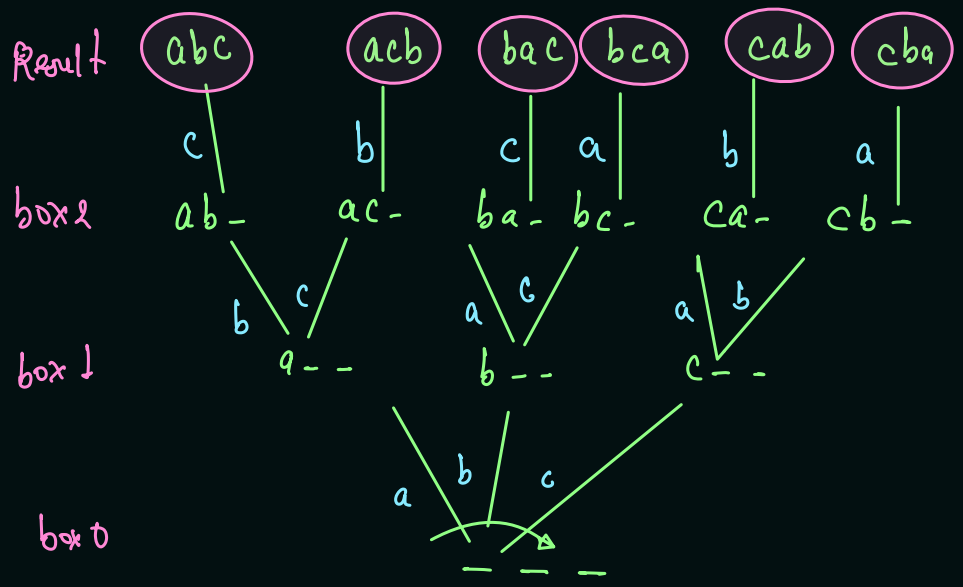
→ box → 3

3-items.

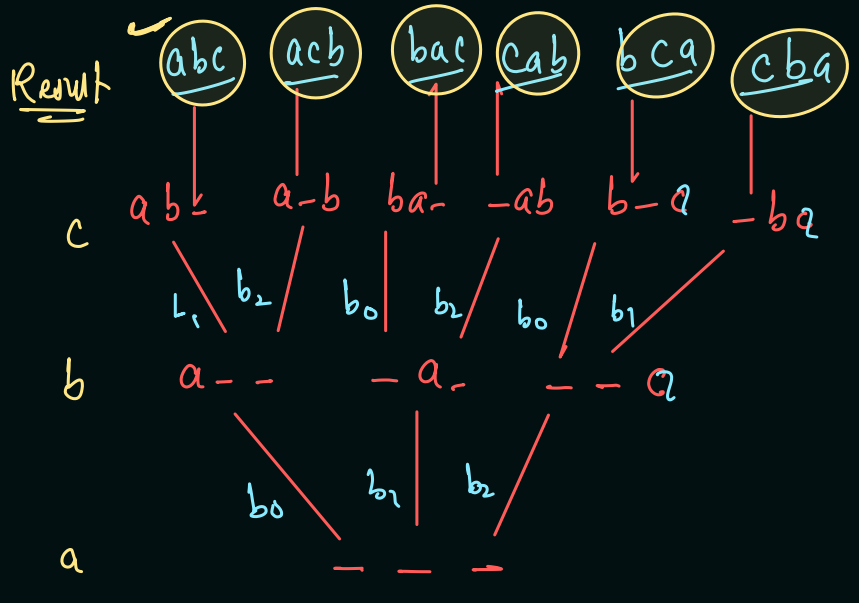
print all permutation to place 3 (non identical) items.

${}^nP_r = {}^3P_3 = \frac{3!}{0!} = \frac{3 \times 2 \times 1}{1} = 6$ $\quad 0! = 1$

level → box



level → items , choice → box



Permutation. Words - 1 , Given a string, print all possible unique arrangement of character.

String = aabbb .

No. of permutation is = $\frac{(\text{Total no. of character})!}{(\text{Repitited character}_1)!\ (\ )_{i_2}!\ \ldots}$

$= \frac{5!}{2!\ 3!} = \frac{5 \times 4^{2} \times 3!}{2! \times 3!} = 10$ possible

$a_1 a_2 a_3$ → permutation : $n! = 3! = 6$

If all characters are same → aaa

$a_1 a_2 a_3$	= aaa
$a_1 a_3 a_2$	= aaa
$a_2 a_1 a_3$	= "
$a_2 a_3 a_1$	= "
$a_3 a_1 a_2$	= "
$a_3 a_2 a_1$	= "

Repitited permutation.

$= \frac{3!}{3!} = 1$

How to avoid Repition of permutation,

for a, for b.

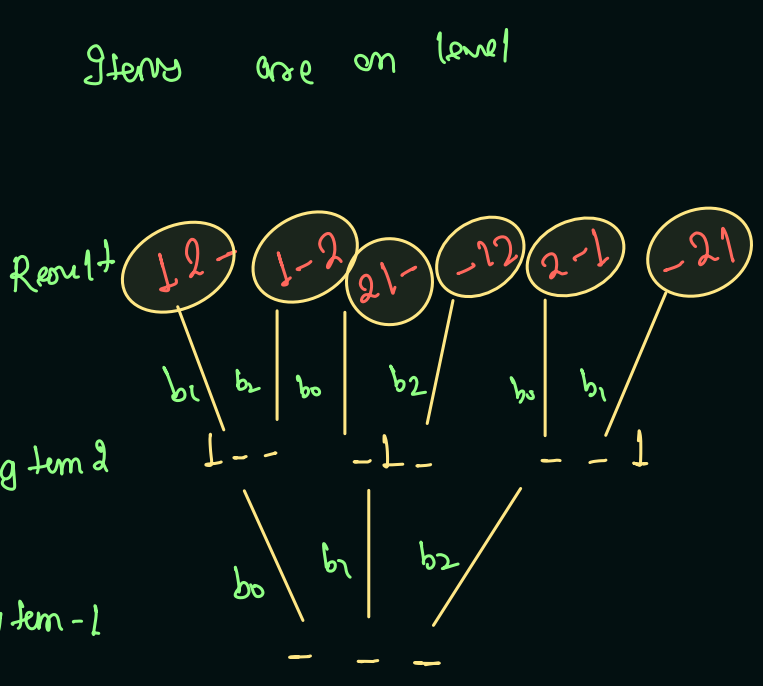
3-box . 2 distinct items (1, 2)

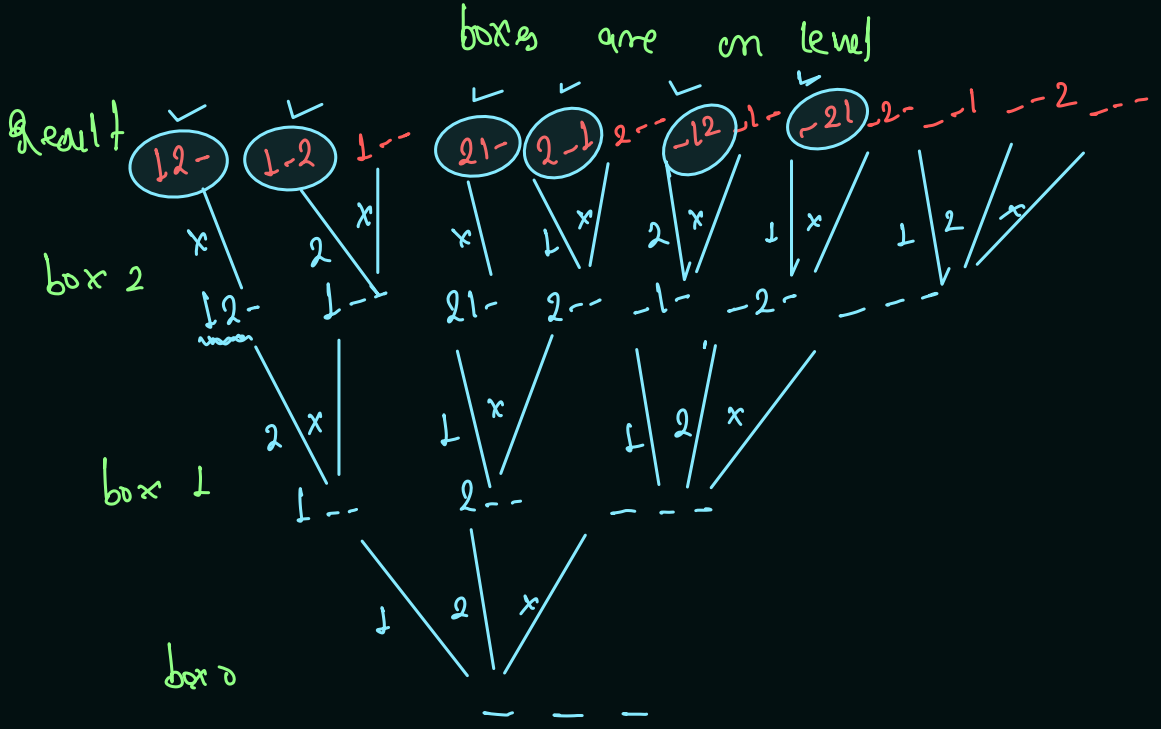
${}^{3}P_2 = \frac{3!}{1!} = 3 \times 2 \times 1 = 6$

ii -	→ 12 -	21 -
i - i	→ 1 - 2	2 - 1
- ii	→ - 12	- 21

Different approaches to Solve permutation →

3 box, 2 items.





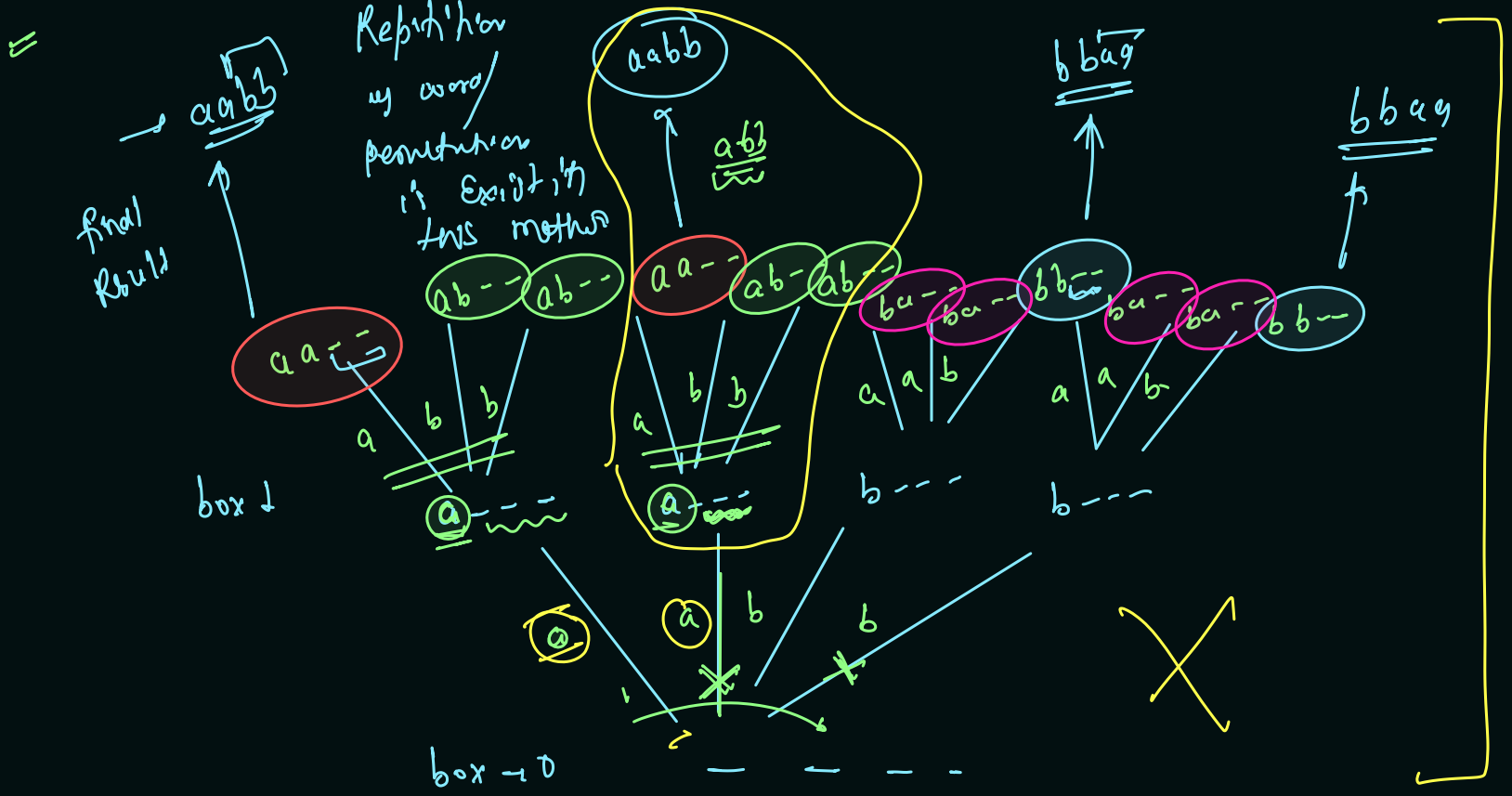
we have to Solve →

a a b b ⇒ total arrangent is : $\frac{4!}{2! \times 2!} = \frac{\cancel{4} \times 3 \times \cancel{2!}}{\cancel{2!} \times \cancel{2}} = 6$

String = aabb

level - box → 4 box ← 4 character (items)

a box can't deny for a character, i.e. it have no option for no calc



avoid this method, here we encounter with Repetition of permutation

aaaaaa → 6!

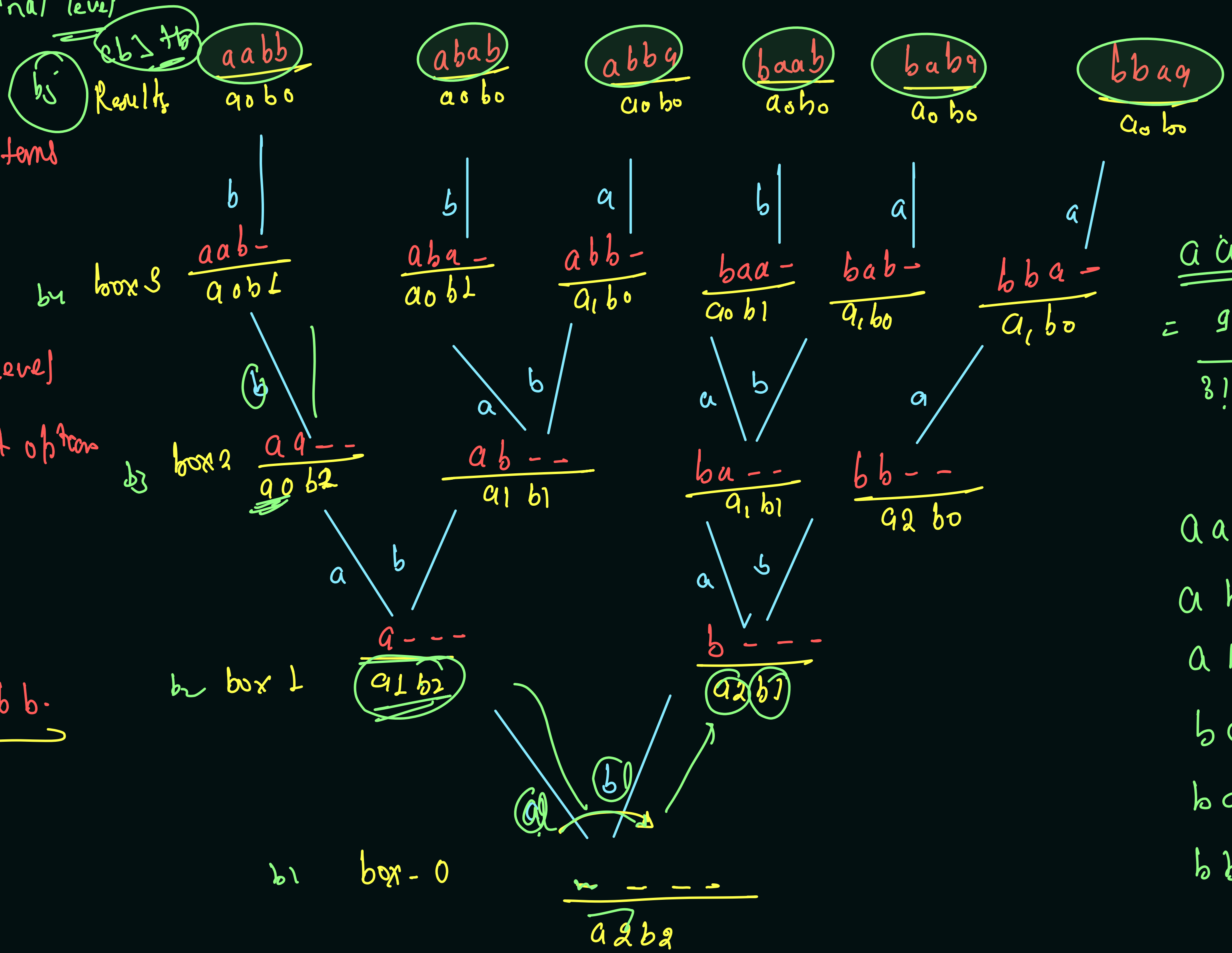
↳ is single final level

4- Character/ items

4- box.

boxes are at level

characters are at option



string → aabb.

freq → a → 2

b → 2

9!

aaabbbccc

$= \frac{9!}{3! \times 3! \times 3!} = \frac{9 \times 8 \times 7 \times 6 \times 5 \times 4}{3 \times 2 \times 3 \times 2}$

9 × 8 × 7 × 10

aabb

abab

abba

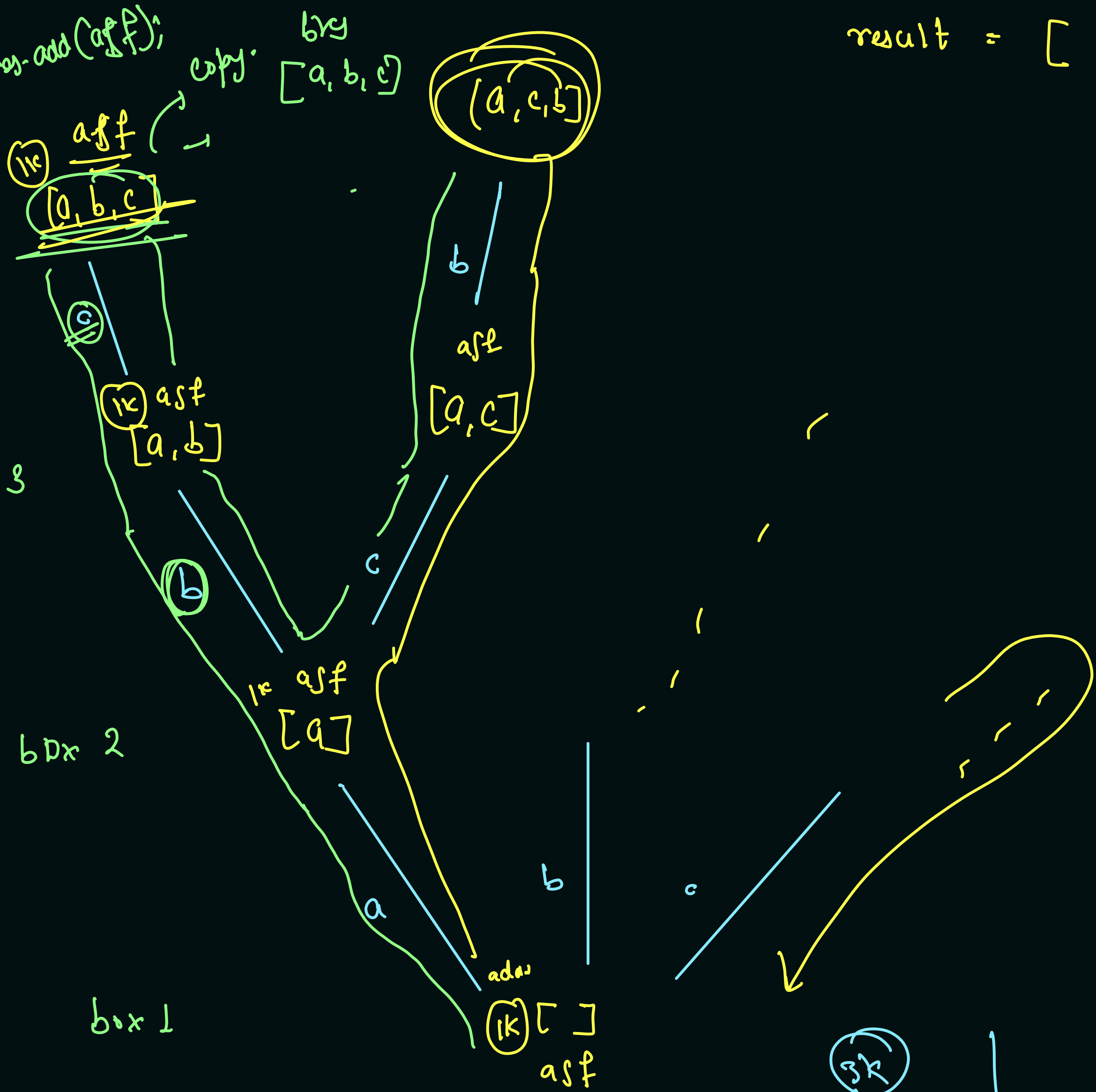
baab

baba

bbaa

character → a, b, c

res.add(asf);

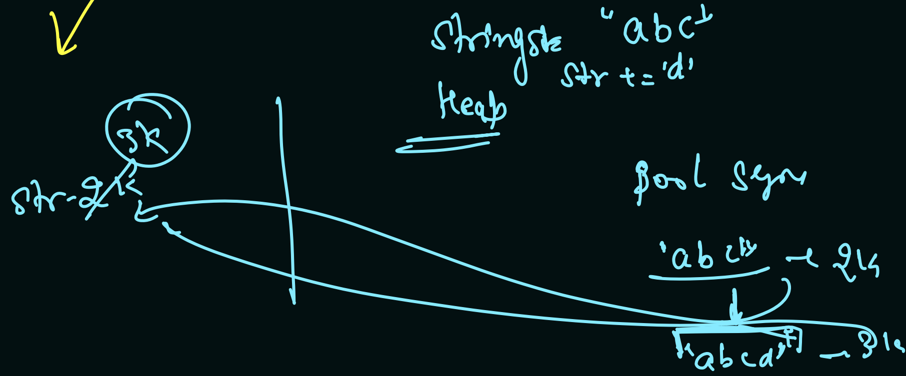


result = [[ ] , [ ]]
asf → 1k
[ ] [ ] [ ]

At base case → make a copy of asf and add it in final result.

Head
First
JAVA

append in String → T.C = O(n)



Permutation _ word . 2

Level → characters

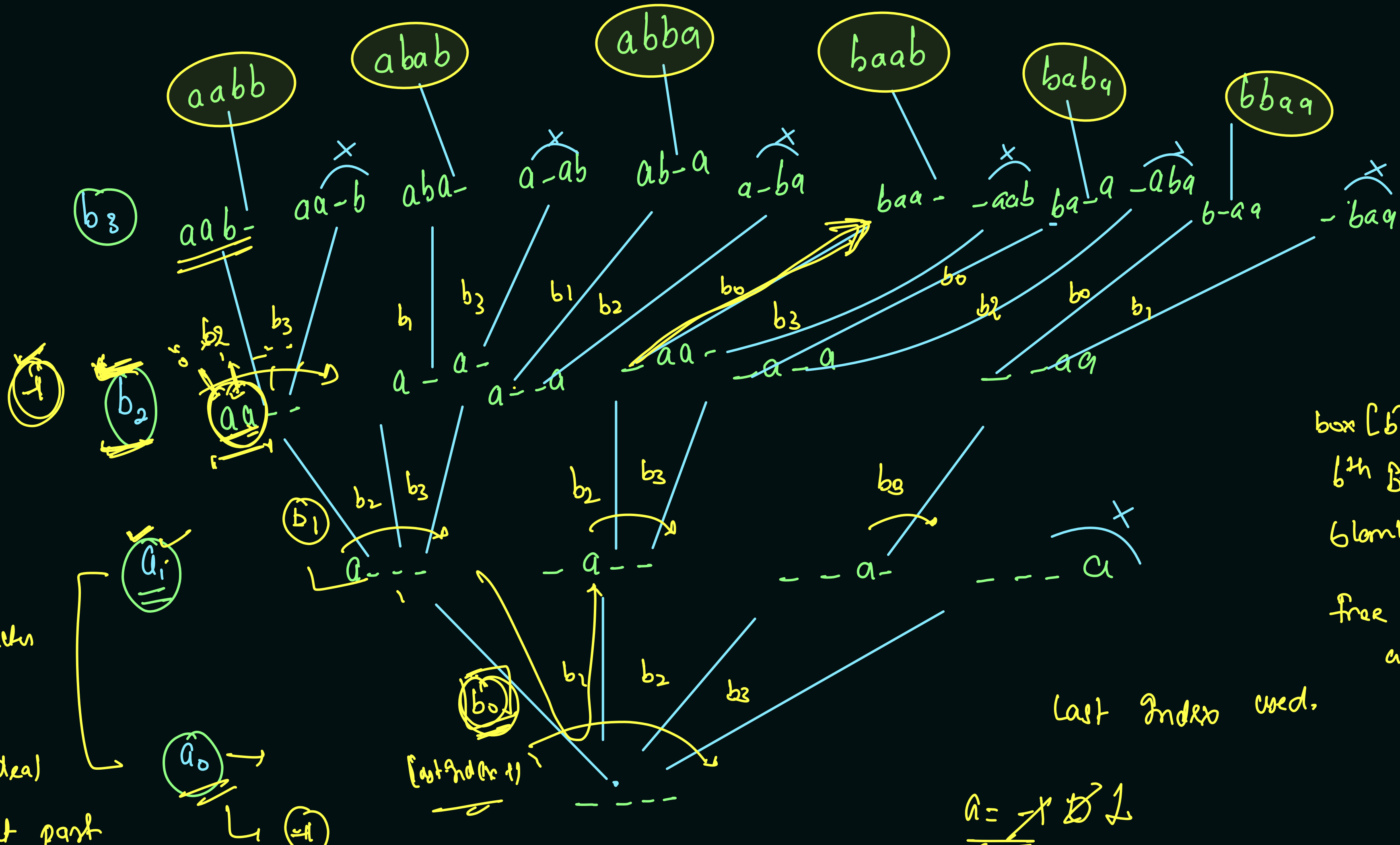
options → boxes.

String → $a_0 a_1 b_2 b_3$

print all possible arrangements of characters which are unique.

[$a_0 a_1 b_0 b_1$
[$a_1 a_0 b_0 b_1$

no. of possibility : $\frac{4!}{2!\,2!} = \frac{4 \times 3 \times 2 \times 1}{4} = 6$



box[b] == null
b^{th} box is blank / empty /
free to place a character

If character is same the one deal with that part as combination