Permutation-Words-1

Sunday, 18 July 2021 9:59 AM

Given a string, all possible unique arrangement of character.

string → aabb $= \dfrac{4!}{\boxed{2!}\boxed{2!}} = \dfrac{\cancel{24} \times 3 \times \cancel{2!}}{\cancel{2} \times \cancel{2}} = \boxed{6}$

3 box, 2 distinct item (1, 2)

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

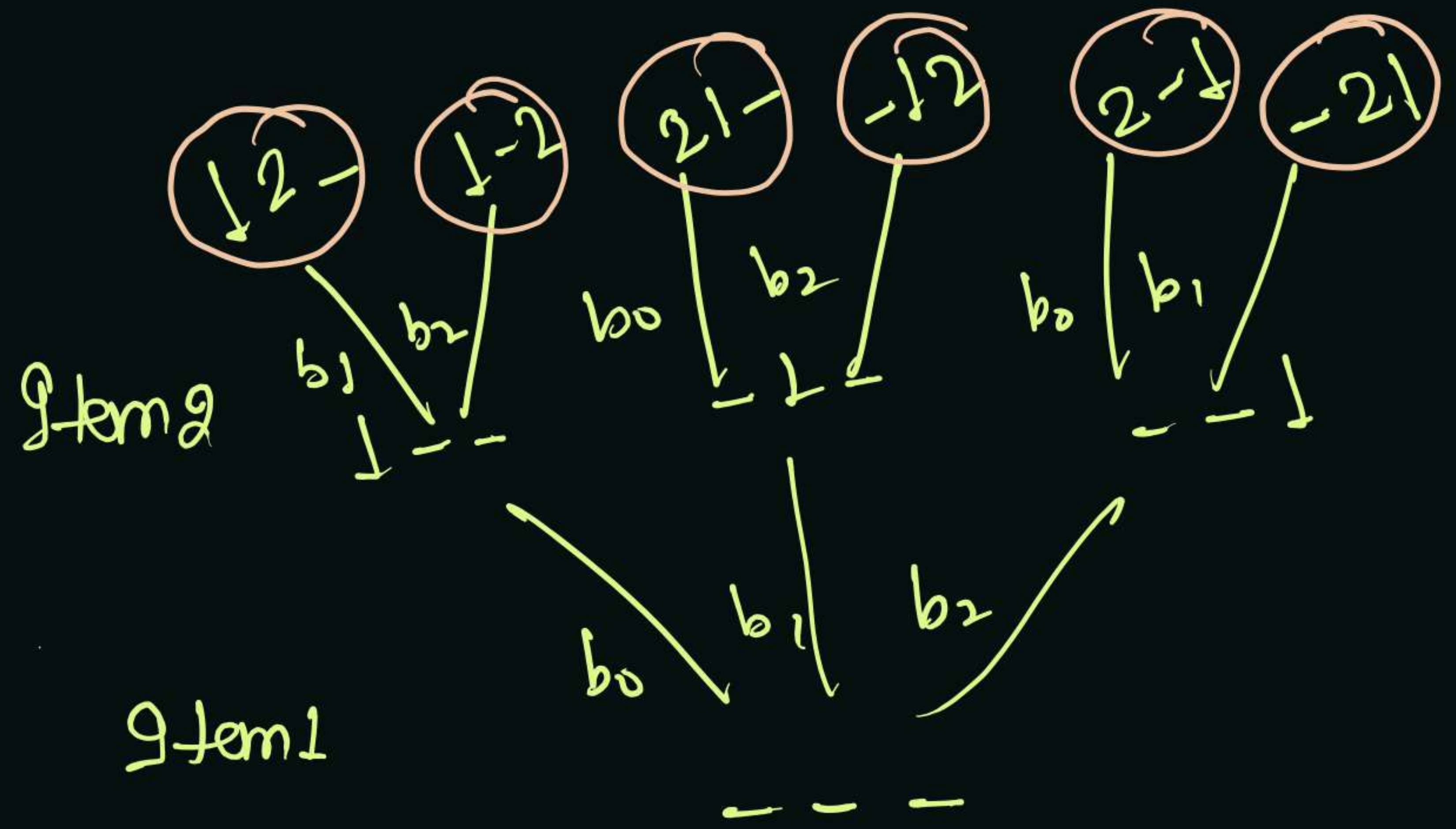
$$^nP_r = (n) \times (n-1)(n-2) \_ \_ \_ (n-(r-1))$$

1 2 _	2 1 _
1 _ 2	2 _ 1
_ 1 2	_ 2 1

Different Approach of permutation :→

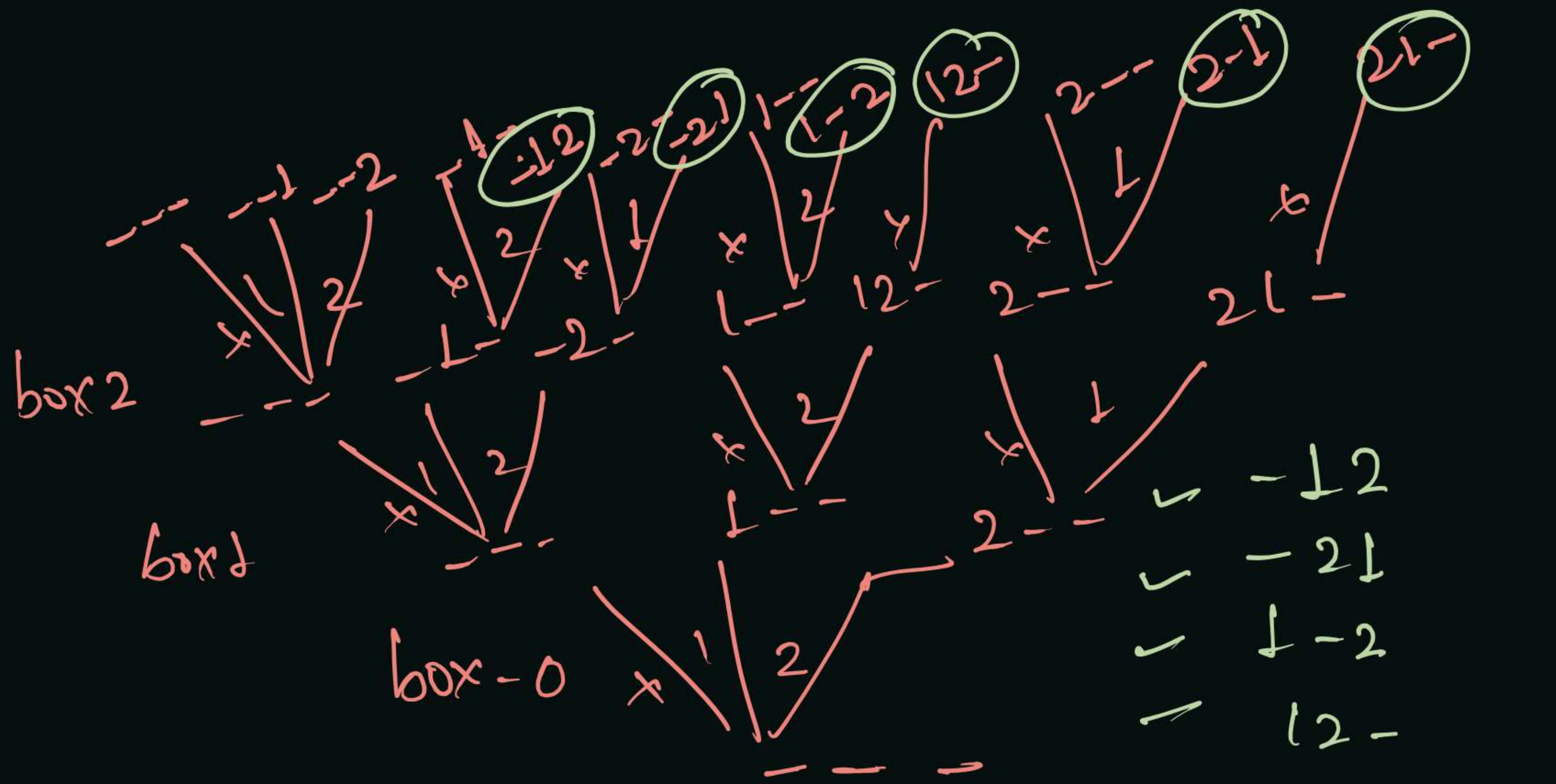
3 box, 2 Items.

Items on level



boxes on level:

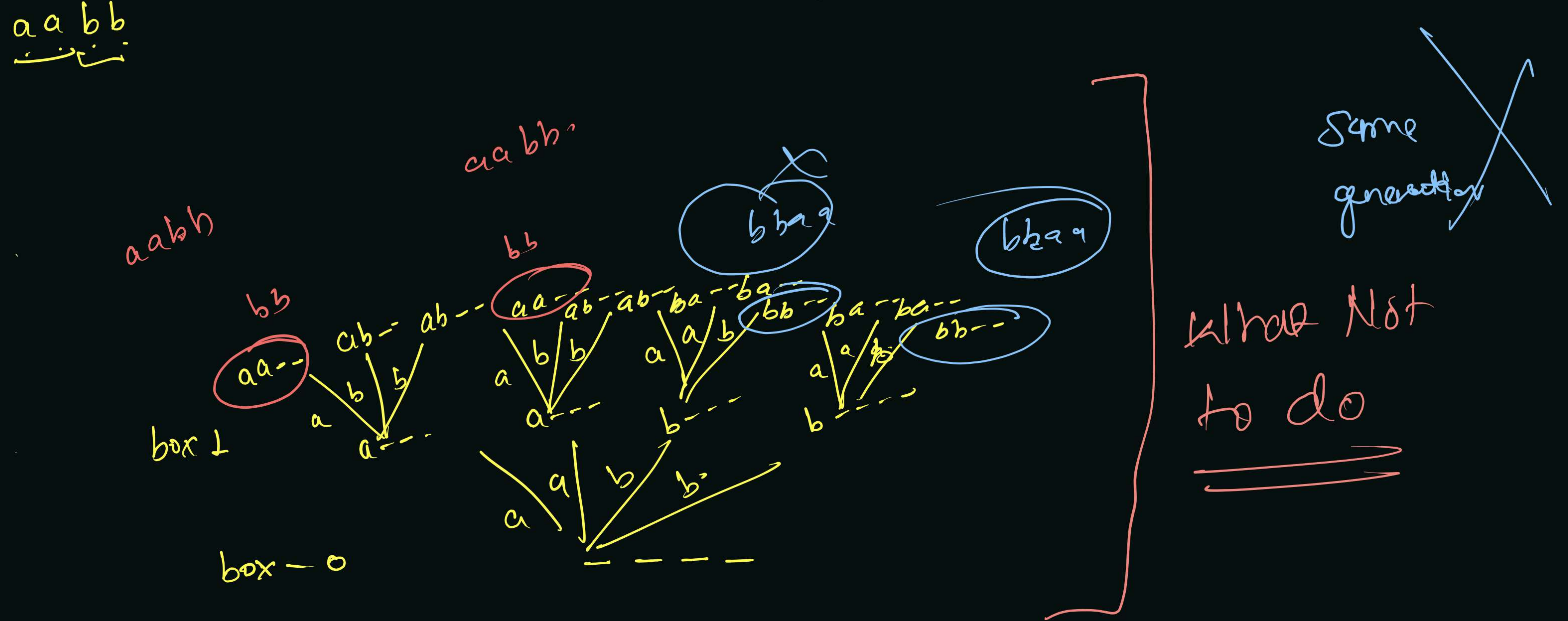
(If box is more than items and box is at level ↳ skip the item)



✓ -12
✓ -21
✓ 1-2
✓ 12-
✓ 2-1
✓ 21-

String → a a b b

level → boxes   → charatter → 4
                 boxes → 4] here box and character are same so No call is not used here,



string → $\underline{aabb}$] freq. map using HashMap.

cb == Tb → box 4 Base case.

cb > tb

box 1

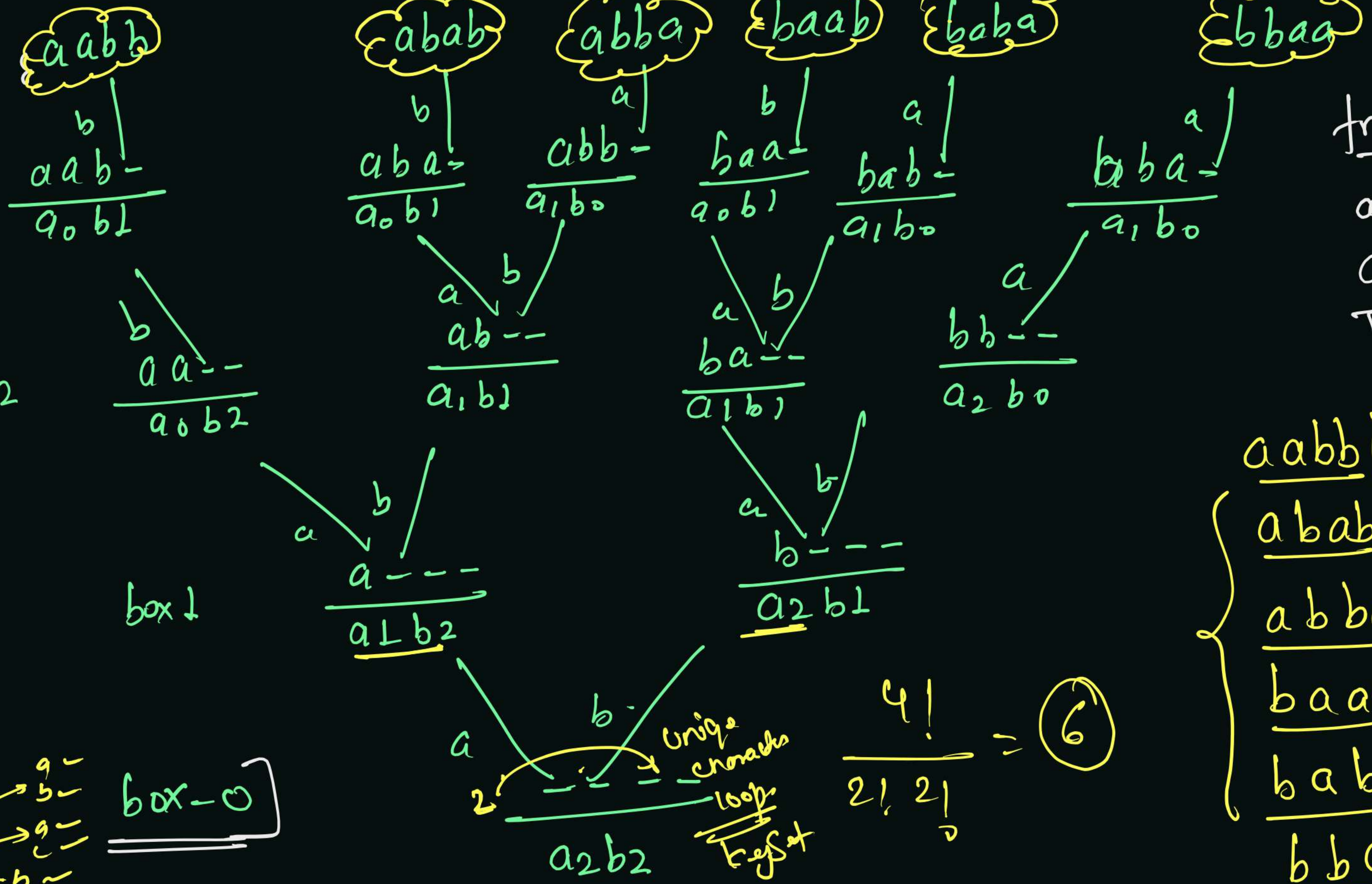
n character.

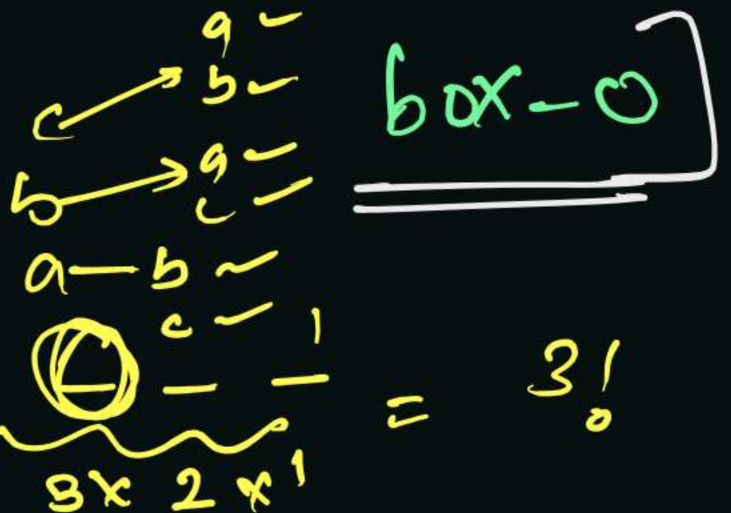
all possible arrangement

↳ n!

Level → boxes.

option → unique available character.





abc

$\underline{3} \; \underline{2} \; \underline{1} = 3!$

$3 \times 2 \times 1$

$n = n \times n-1 \times n-2 \ldots 1 = n!$

$\frac{4!}{2!\,2!} = \boxed{6}$

fmap:

asf

Cb:

Tb:

aabb ✓

abab

abba

baab

baba ✓

bbaa.

Permutation-Words-2

Sunday, 18 July 2021 9:59 AM

3 box, 2 distint items (1, 2)

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

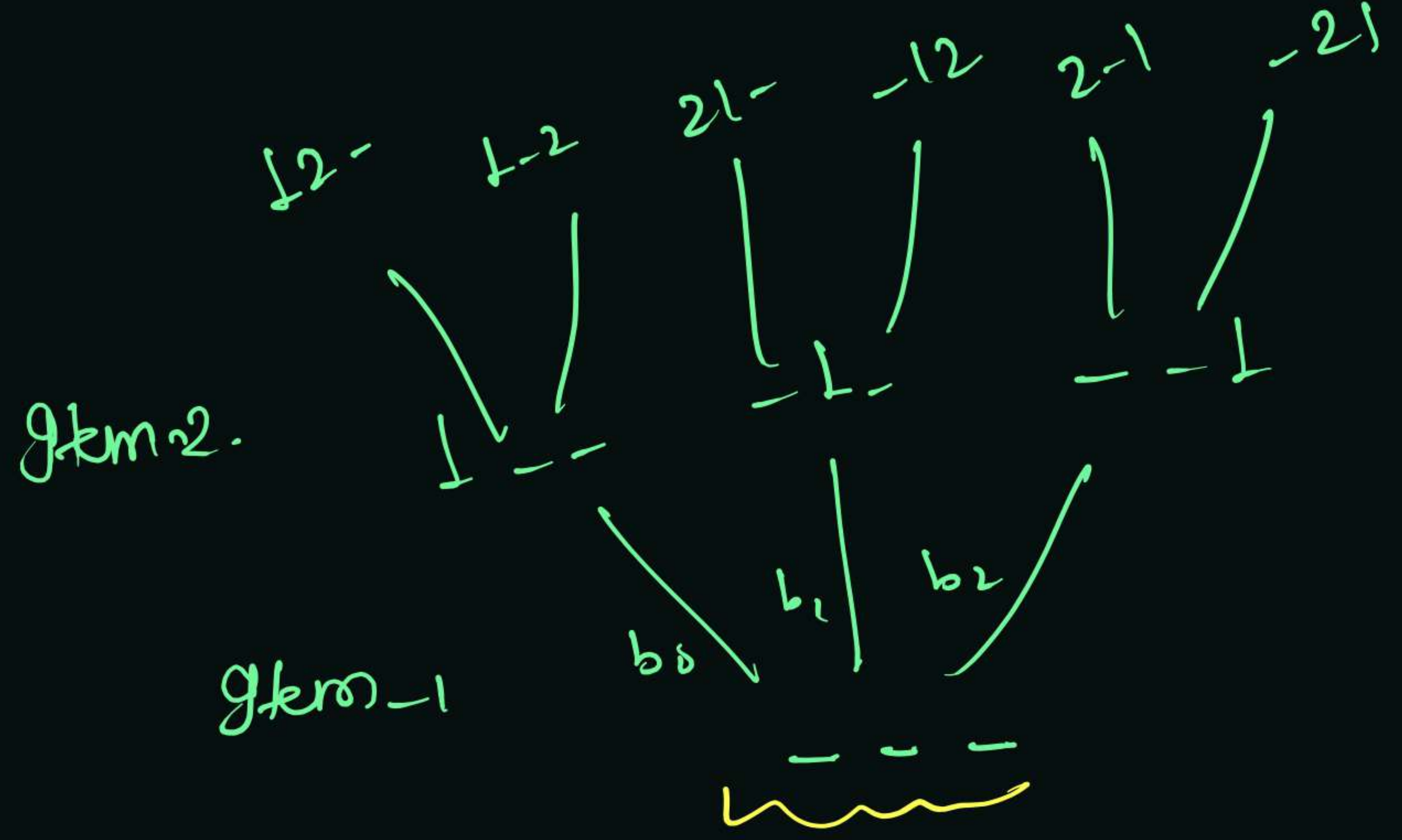
Item : level → character

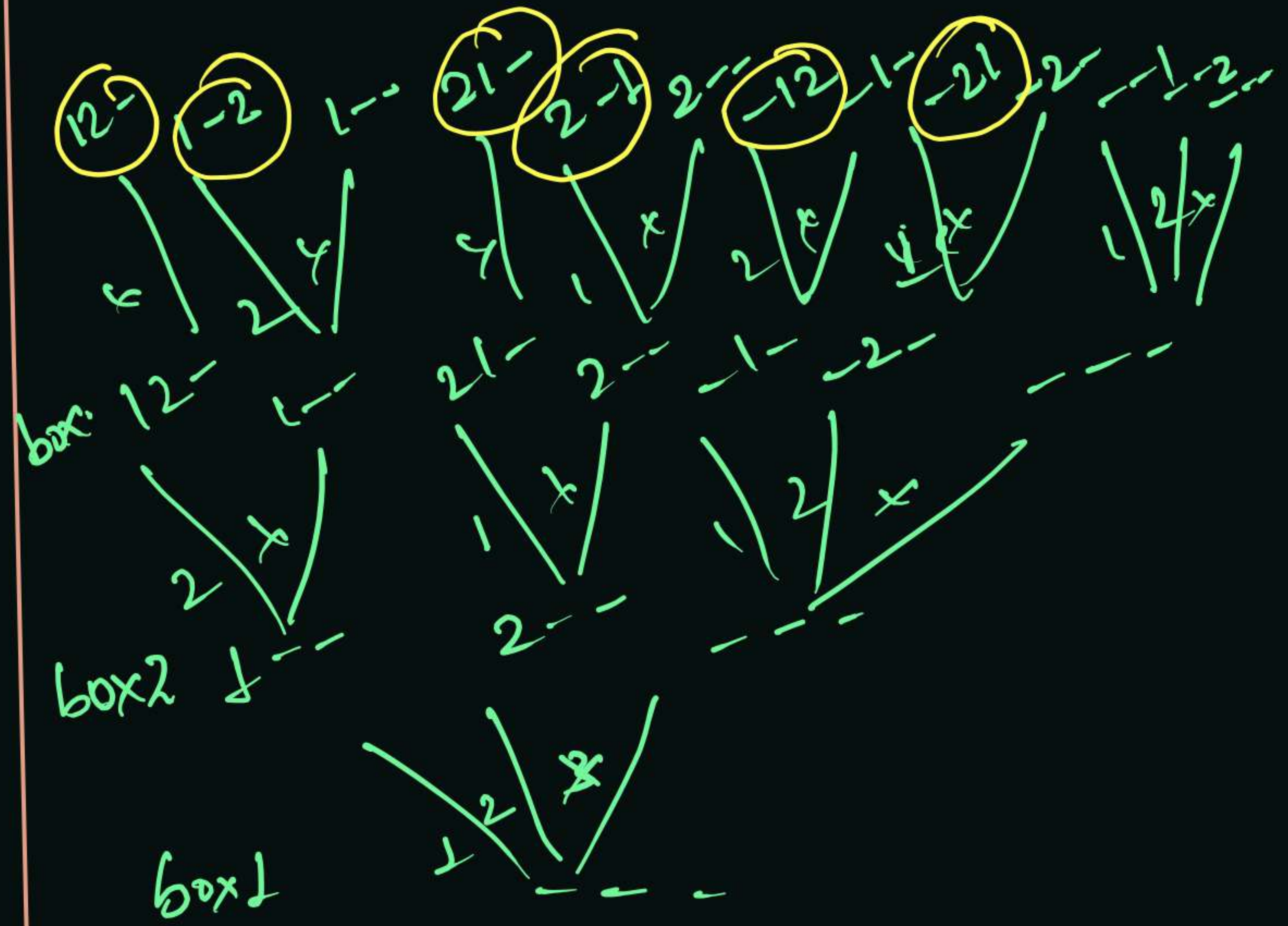
box : level → spaces

1 2 -   2 1 -

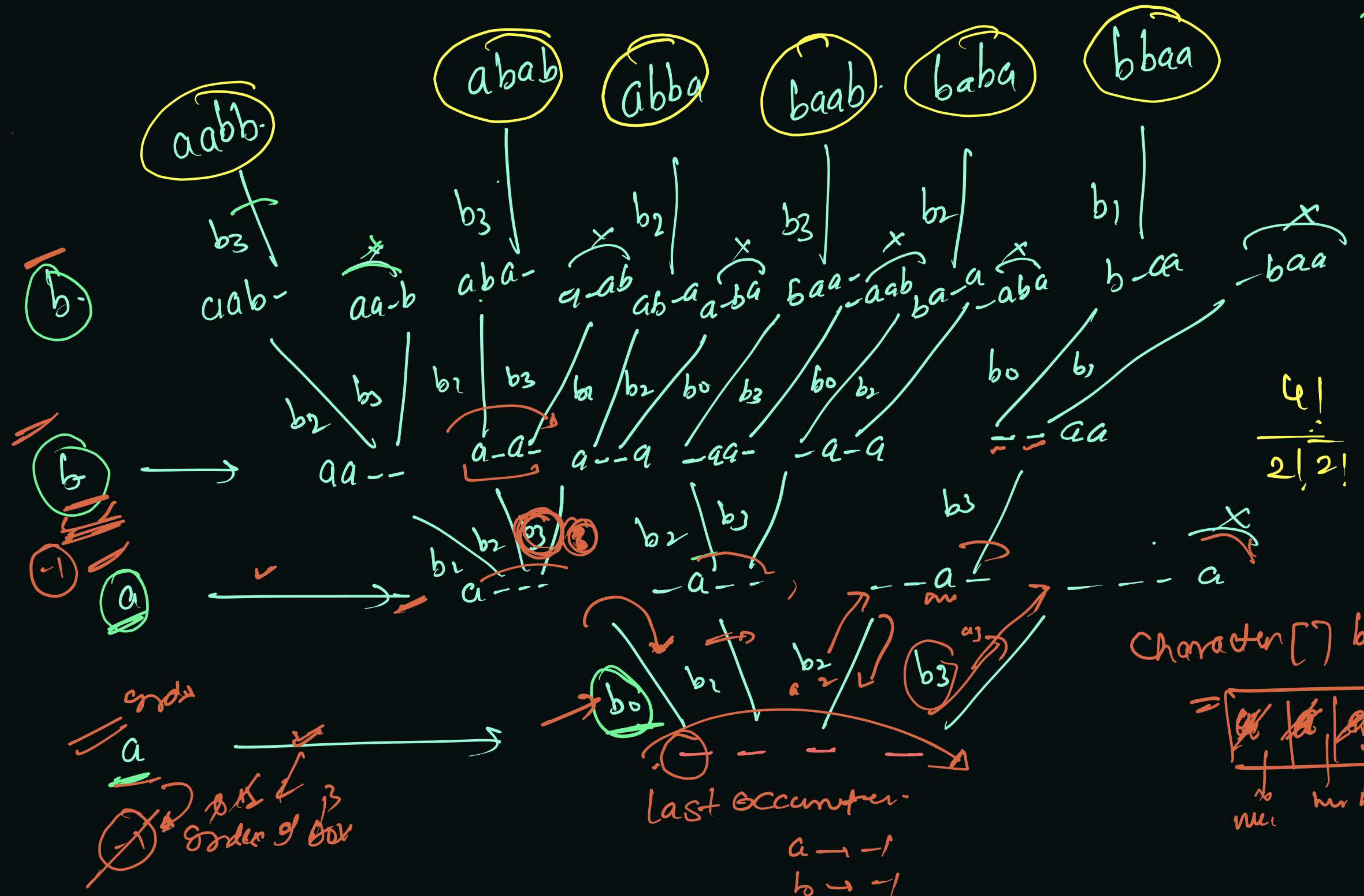
1 - 2   2 - 1

- 1 2   - 2 1





Character → level



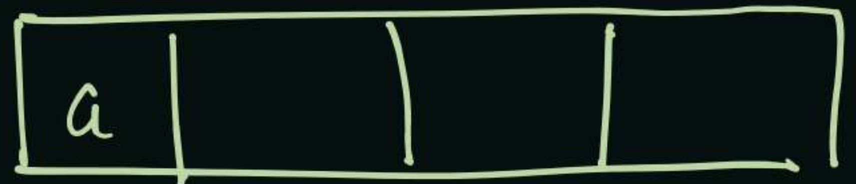
$$\frac{4!}{2!\,2!} = 6$$

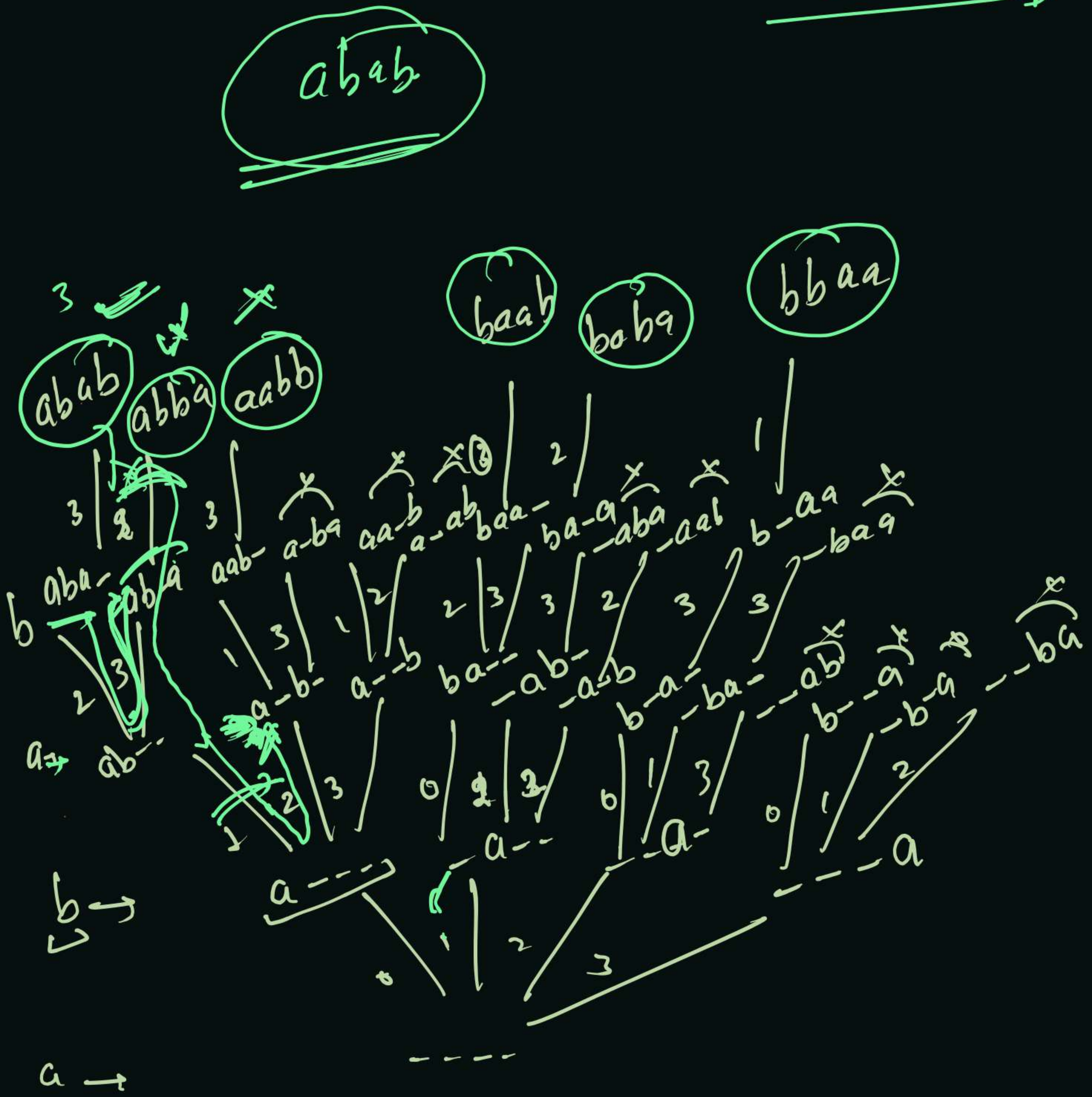
Combination feel] to avoid Repetition we make ahead call

aabb from
abab previous
abba same
baab character
baba selecting
bbaa

String → abab

aabaa →





```
// cc-> current character, li-> last index
public static void generateWords(int cc, String str, Character[] spots,
    if(cc == str.length()) {
        for(char c : spots) {
            System.out.print(c);
        }
        System.out.println();
        return;
    }

    char ch = str.charAt(cc);
    int lsi = li.get(ch); // last spot index

    for(int box = lsi + 1; box < spots.length; box++) {
        if(spots[box] == null) {
            spots[box] = ch;
            li.put(ch, box);
            generateWords(cc + 1, str, spots, li);
            spots[box] = null;
            li.put(ch, lsi);
        }
    }
}
```

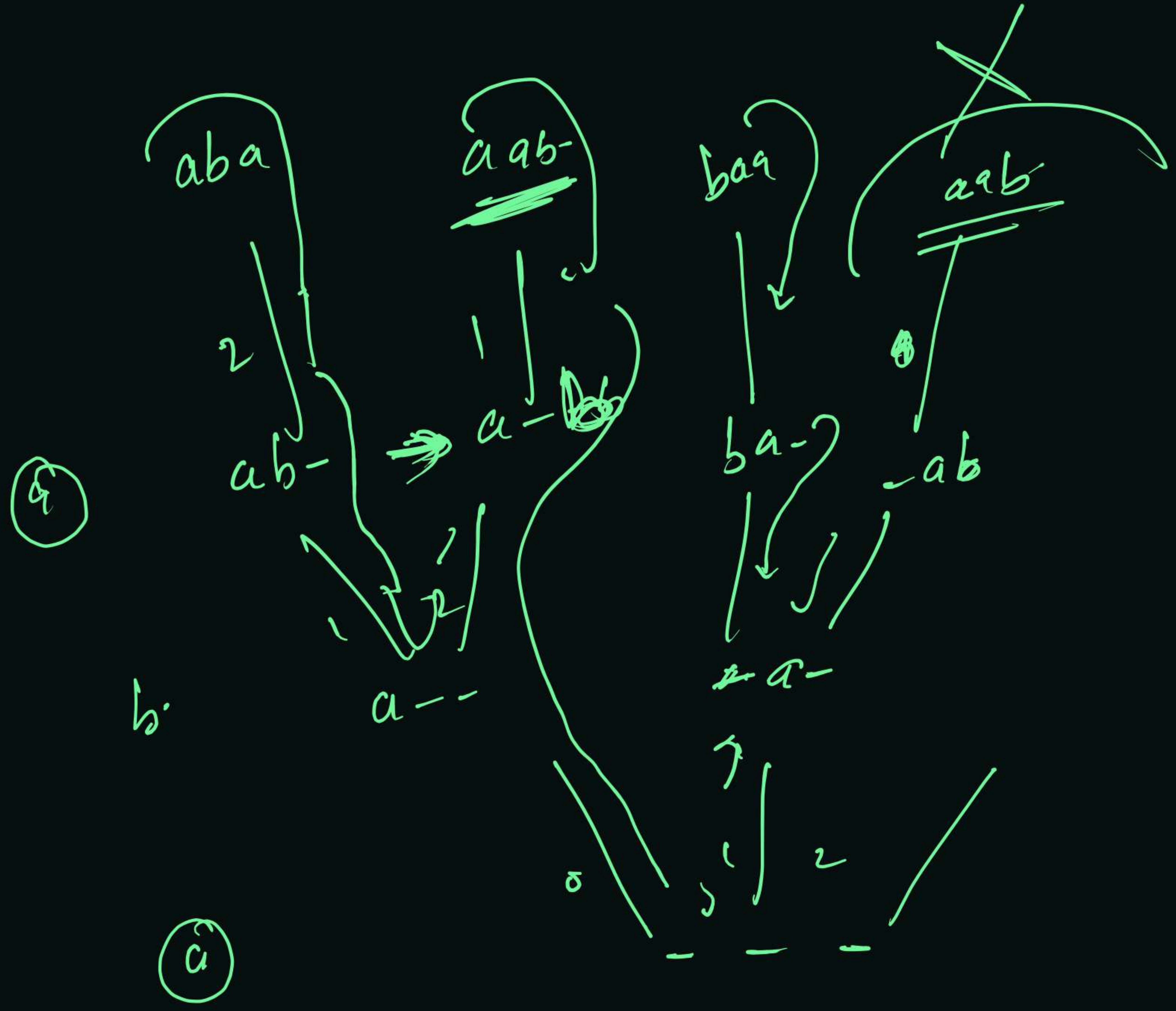
aba.

Why last index is required to set of

lsi ?

$a \to \cancel{1}\, 0$

$b \to 2$



aba
aab
baa

Words-K-Selection-1

Sunday, 18 July 2021 9:59 AM

Given a string, you have to pick k distinct character and print them. (NOTE: Not arrange picked character)

Ex. String str = aab.

K=2 C ab

Not ba, Not aa

3 box, 2 same item (i, i)

$$^3C_2 = \frac{3!}{1!\,2!} = \boxed{3}$$

$$^nC_r = \frac{n!}{(n-r)!\,r!}$$

i i _

i _ i

_ i i

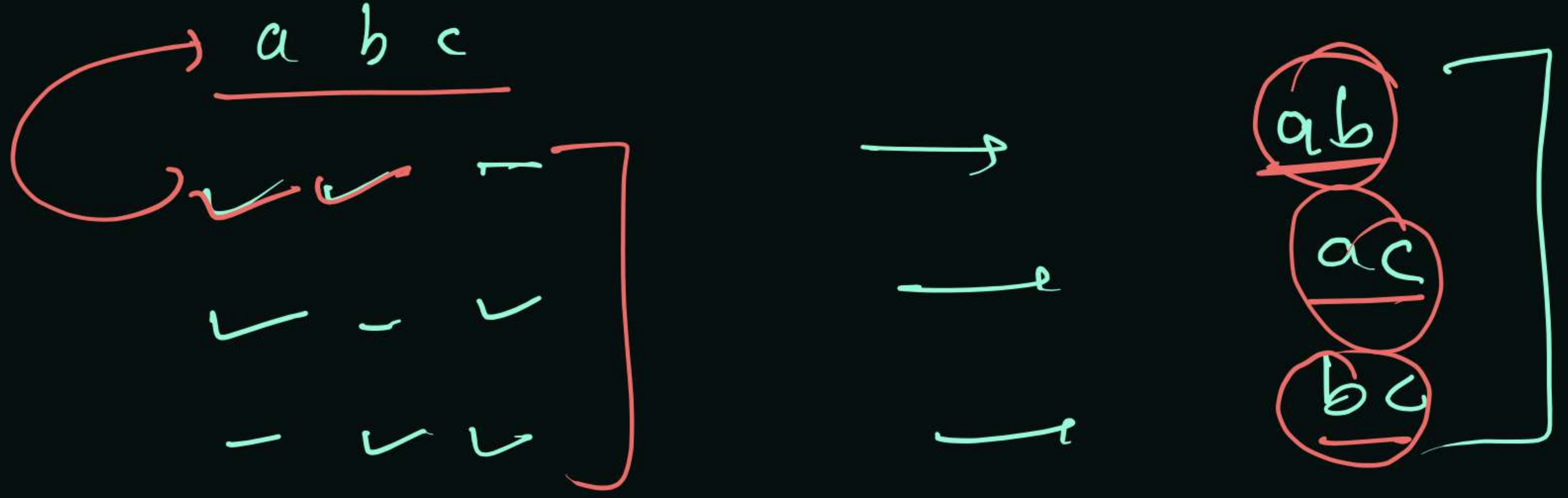
Select 2 box out of 3 box

string → abc abc

unique string → "abc"

select 2 character out of these 3 characters

a b c



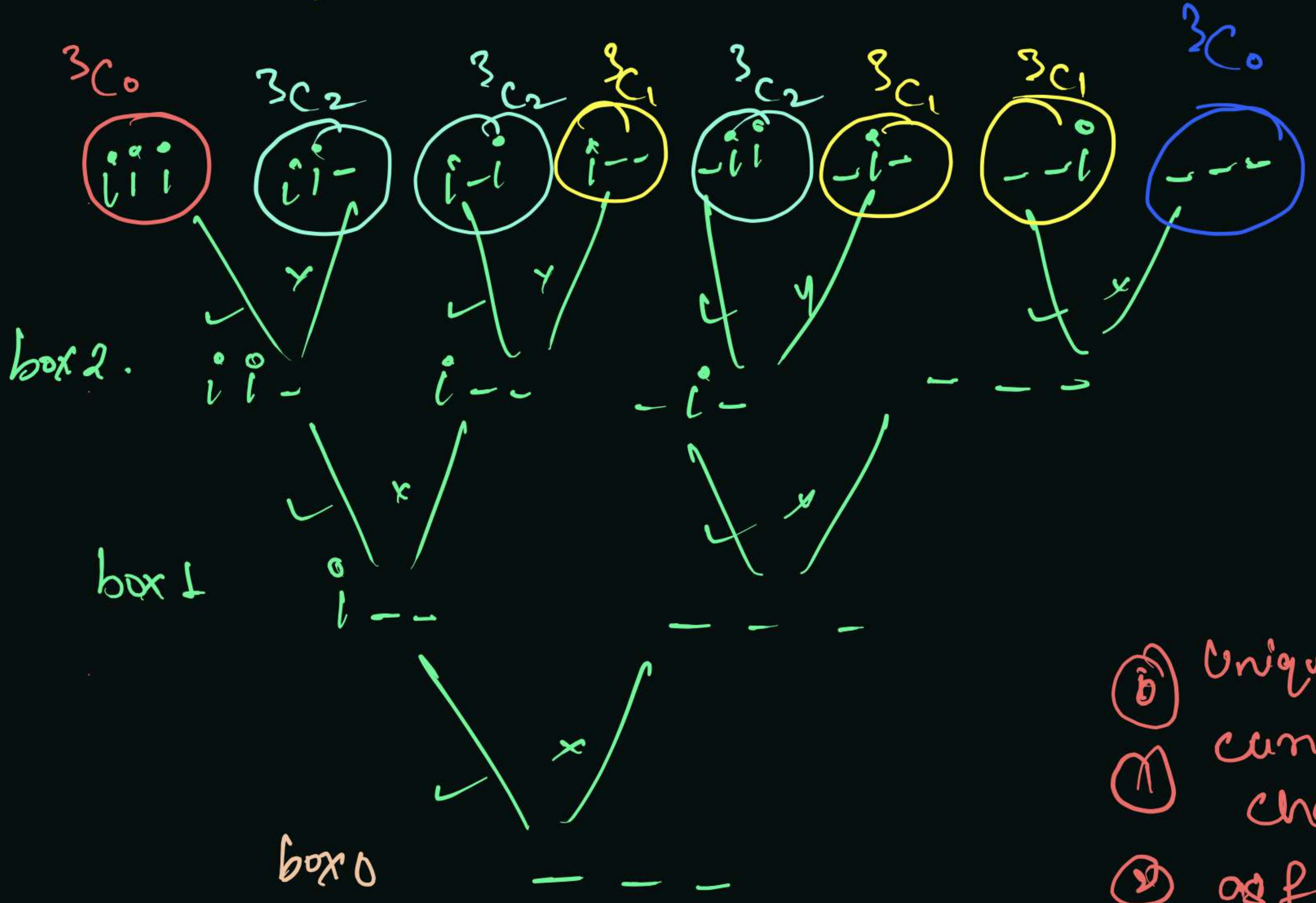
① Generate unique string.

② pick K character out of n unique character (not arrange them) → Becuse of combinat.

combination

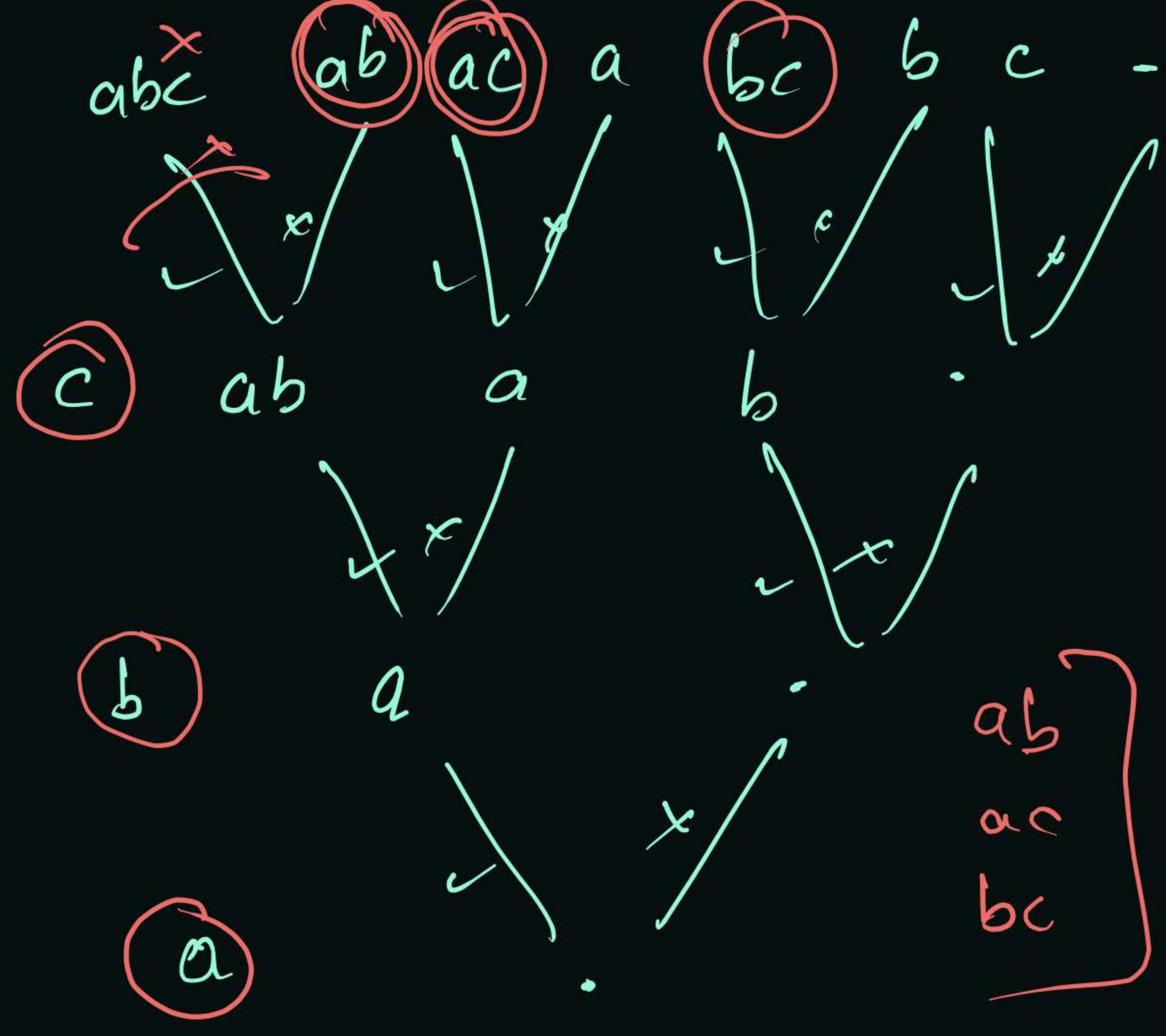
$$2^n = {}^nC_0 + {}^nC_1 + {}^nC_2 + \ ---\ + {}^nC_n$$

box = 3
↳ level. $2^3 = {}^3C_0 + {}^3C_1 + {}^3C_2 + {}^3C_3$



$\boxed{k=2}$ $\underline{a\ b\ c}$] ≡ boxes index

$^3C_2$



ab
ac
bc

⓪ Unique String.
① current character
② asf
③ ssf =>
④ k

Words-K-Selection-2

Sunday, 18 July 2021 9:59 AM

Given string, select k unique charater from String.

3 box, 2 Item (i, i)

$^nC_r = \frac{n!}{(n-r)!\,r!}$

$^3C_2 = \frac{3!}{2!1!} = \boxed{3}$

i i _

i _ i

_ i i

selection of 2 box out of 3 box

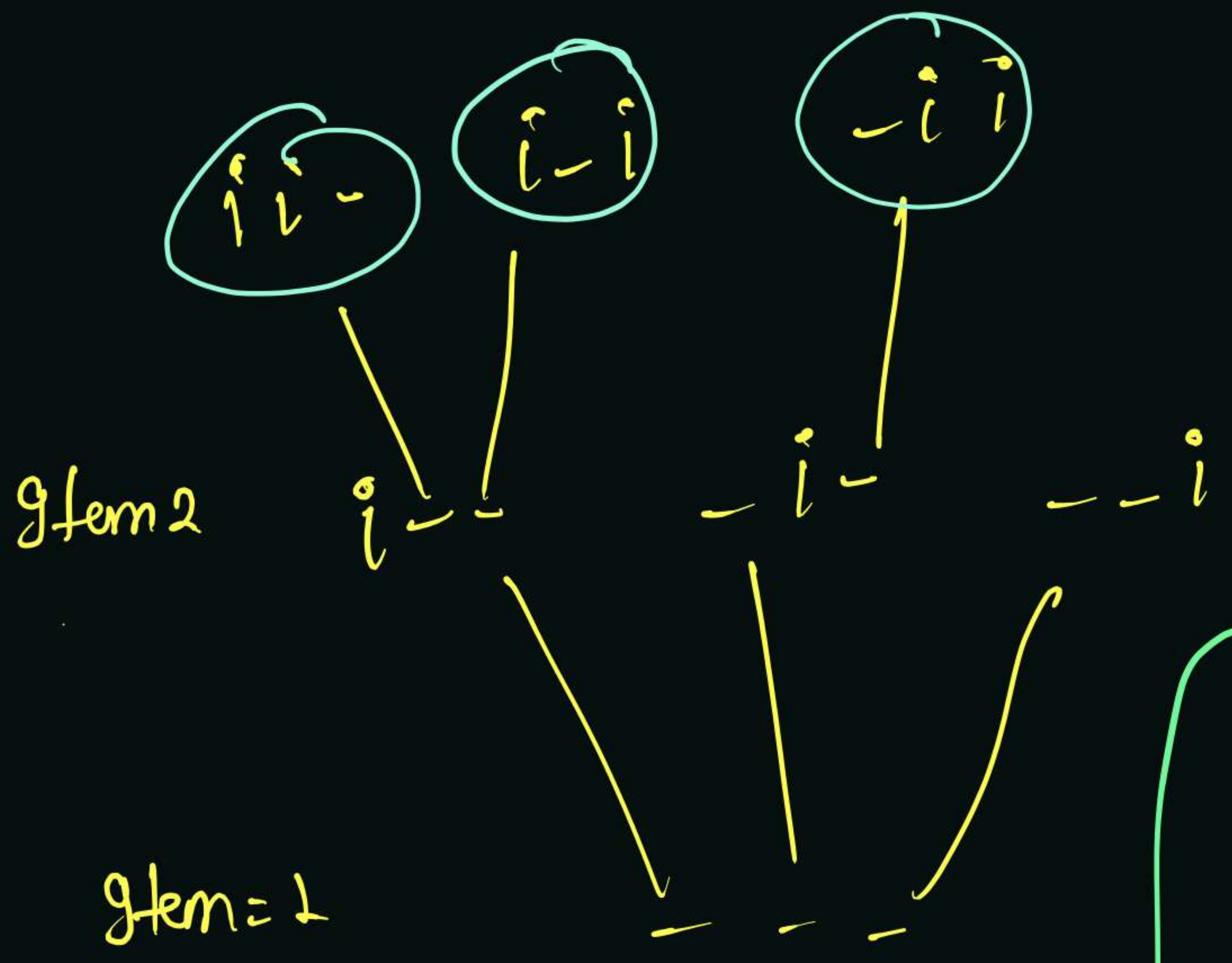
We have string, string - abcabc

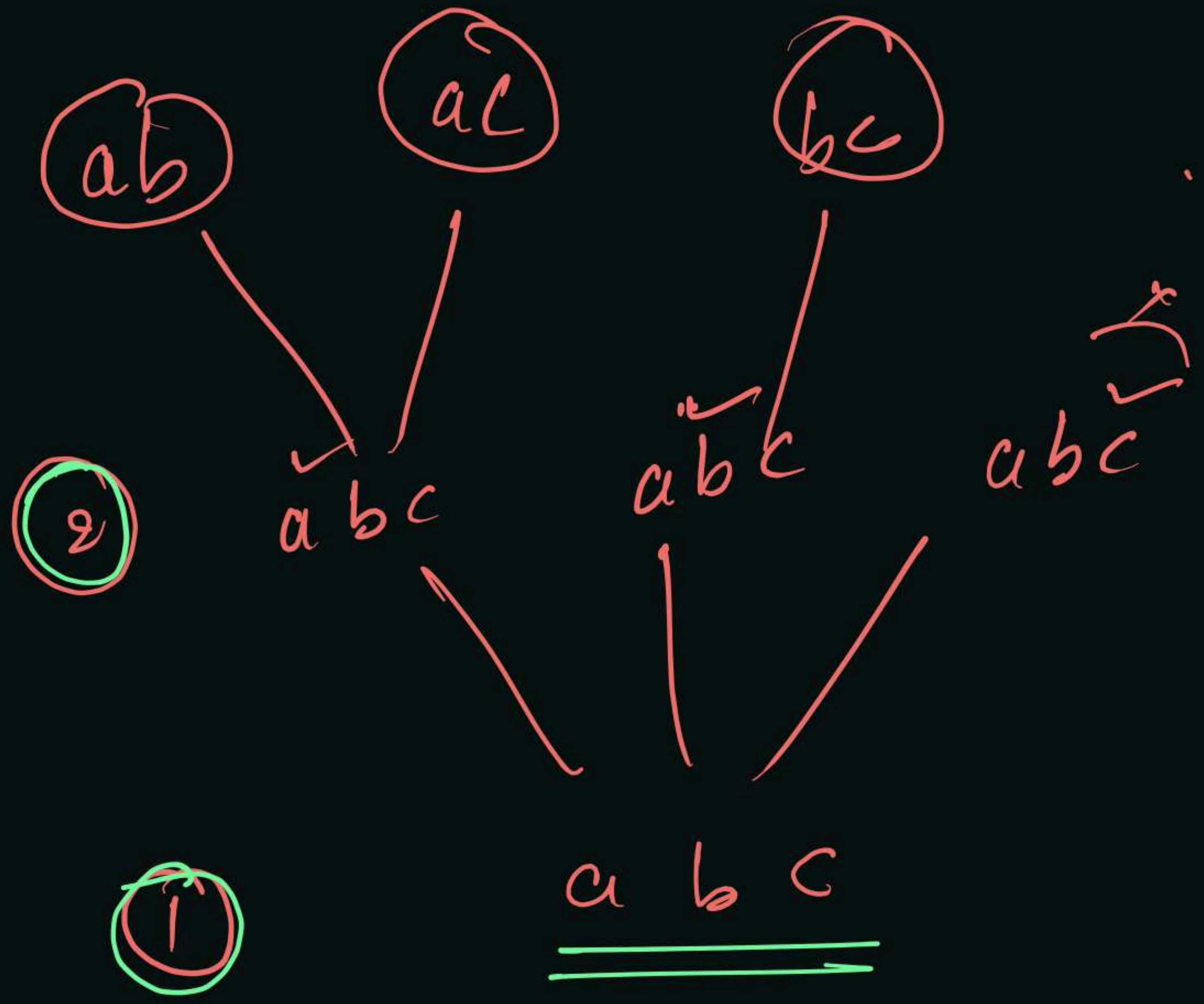
unique → abc

problem select k charcter from unique string.

level - item.

Character selection is on level





string.
ssf
ts
asf
last character

Words-K-Length-Words-1

Sunday, 18 July 2021 9:59 AM

Given a string str, make all possible words of length K with unique character.

string → abcabc , ustr: abc

→ ab ba

ac ca

bc cb

3 boxes, 2 distinct item (1, 2)

1 2 _ 21 _

1 _ 2 2 _ 1

_ 1 2 _ 21

a b c
_ _ _

ab ba

ac ca

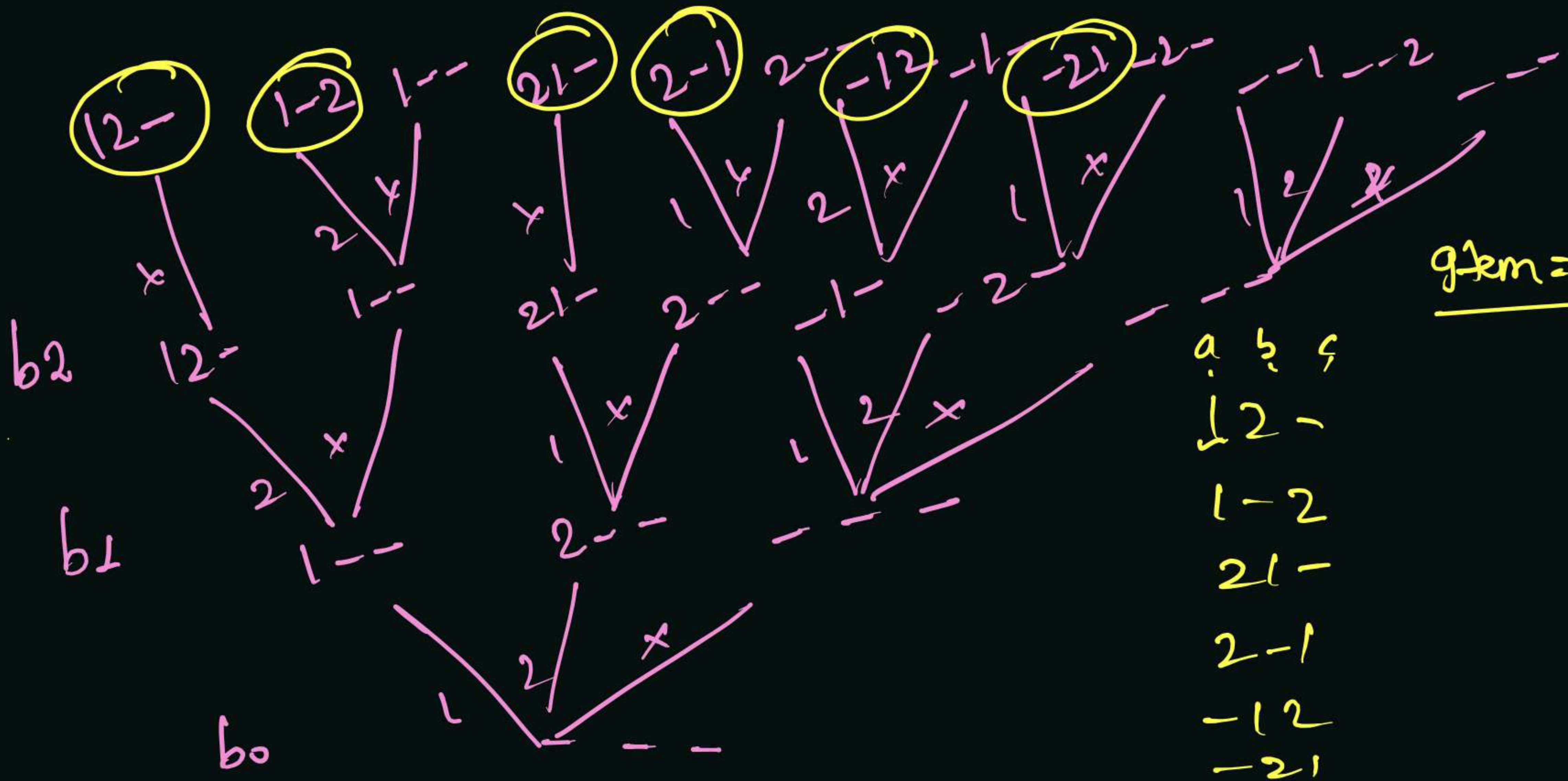
bc cb.

box	Item.	Character	Spots.
b1	i1	a	S1
b2	i2	b	S2
b3		c	

K-spots.

(box-level)

3 box 2 Item.



Item == Slot

a	b	c
1	2	-
1	-	2
2	1	-
2	-	1
-	1	2
-	2	1

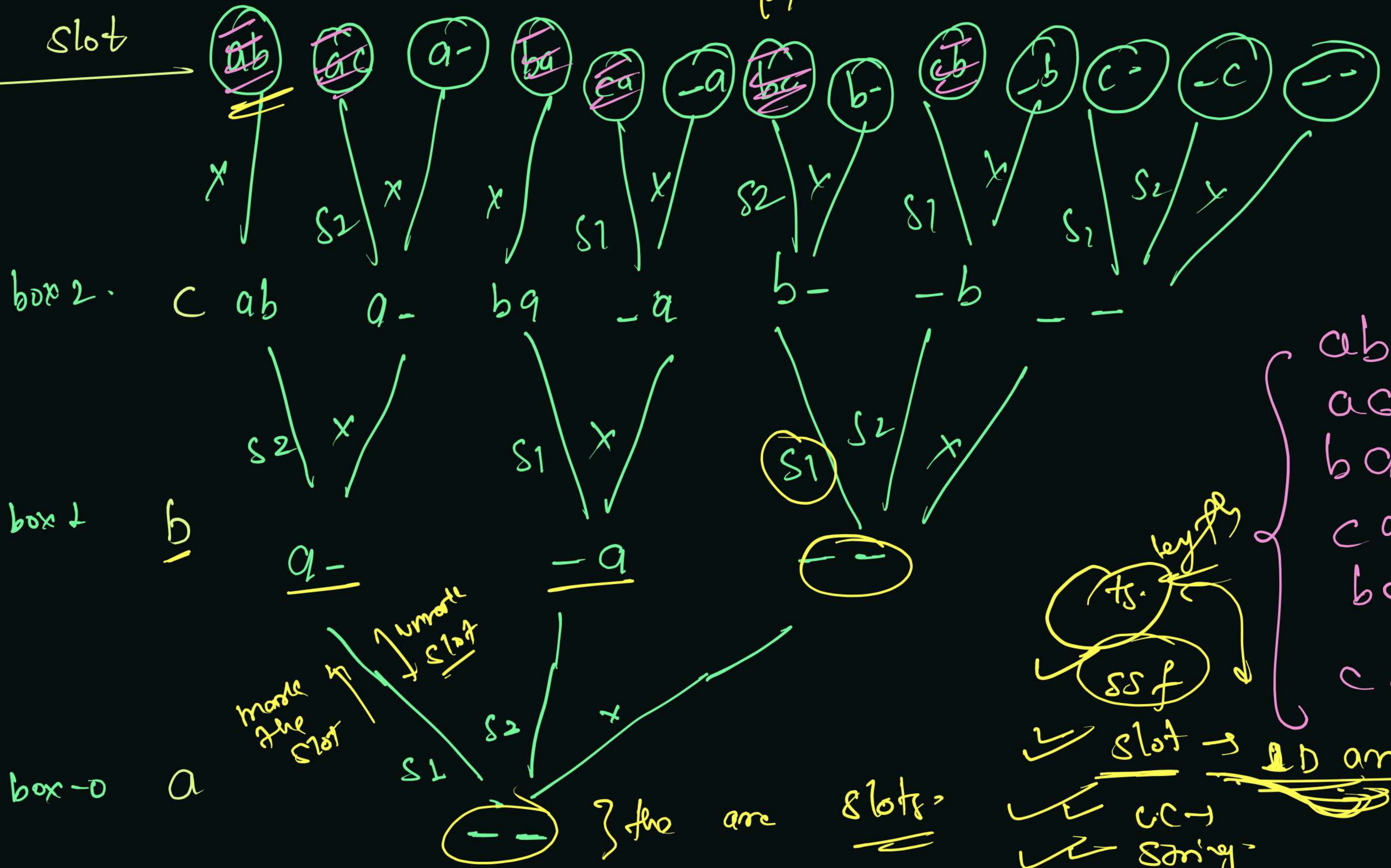
a	b
a	c
b	a
c	a
b	c
c	b

box = abc

item = slot

Boxes are Mapped with character.
Items are Mapped slot

level → character of string



ab
ac
ba
ca
bc
cb

slot → 1D array

c.c →

string