Power log

x^n

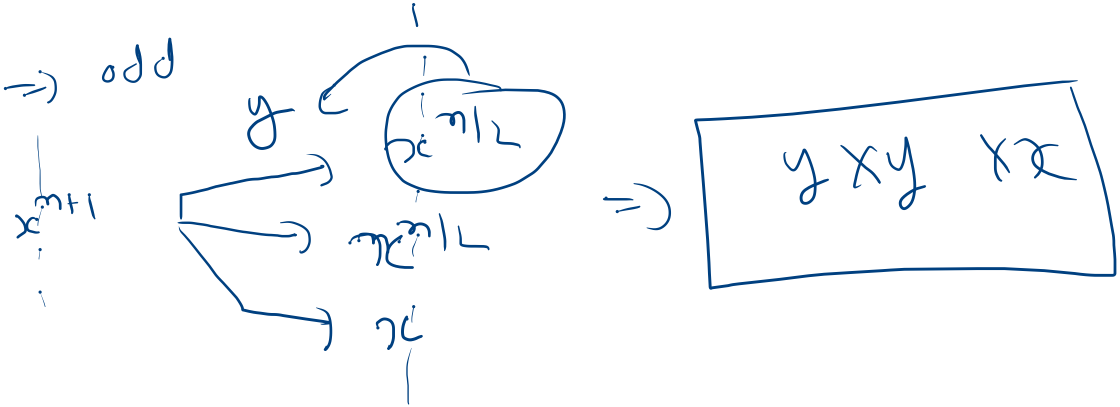
$n = odd$

$x^{n+1} \Rightarrow x \cdot x^n$

$\Rightarrow x \cdot x^{n/2} \cdot x^{n/2}$

2^5

$n+1 \Rightarrow odd$



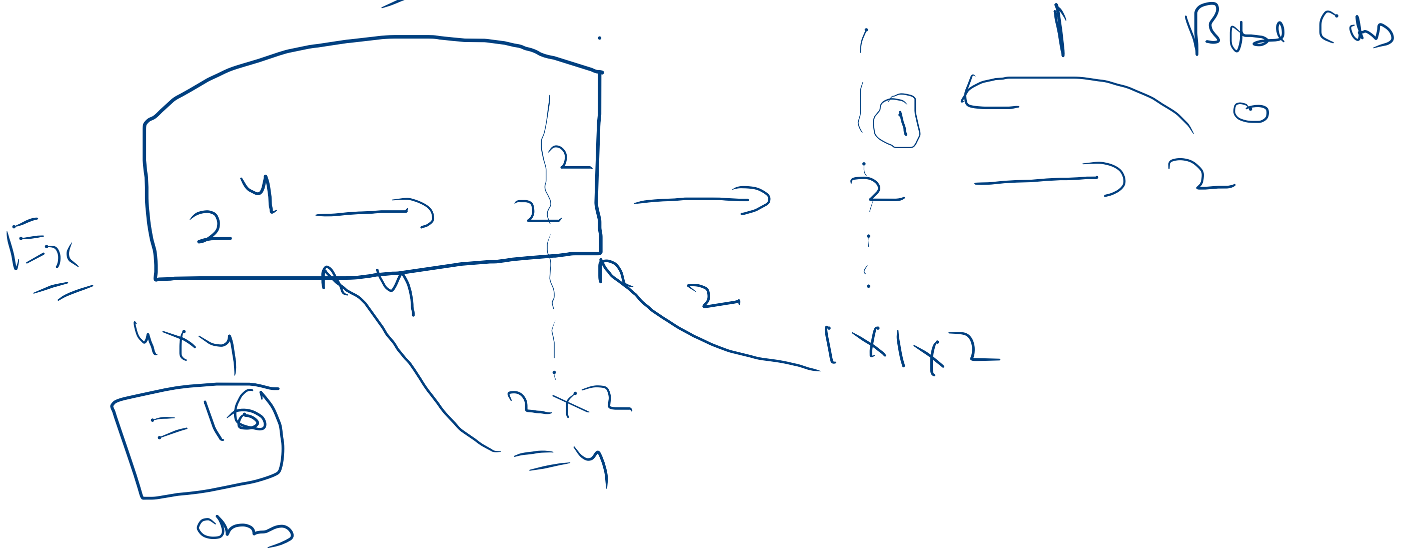
y = recursion ans

$x^n \rightarrow x^{n/2}$

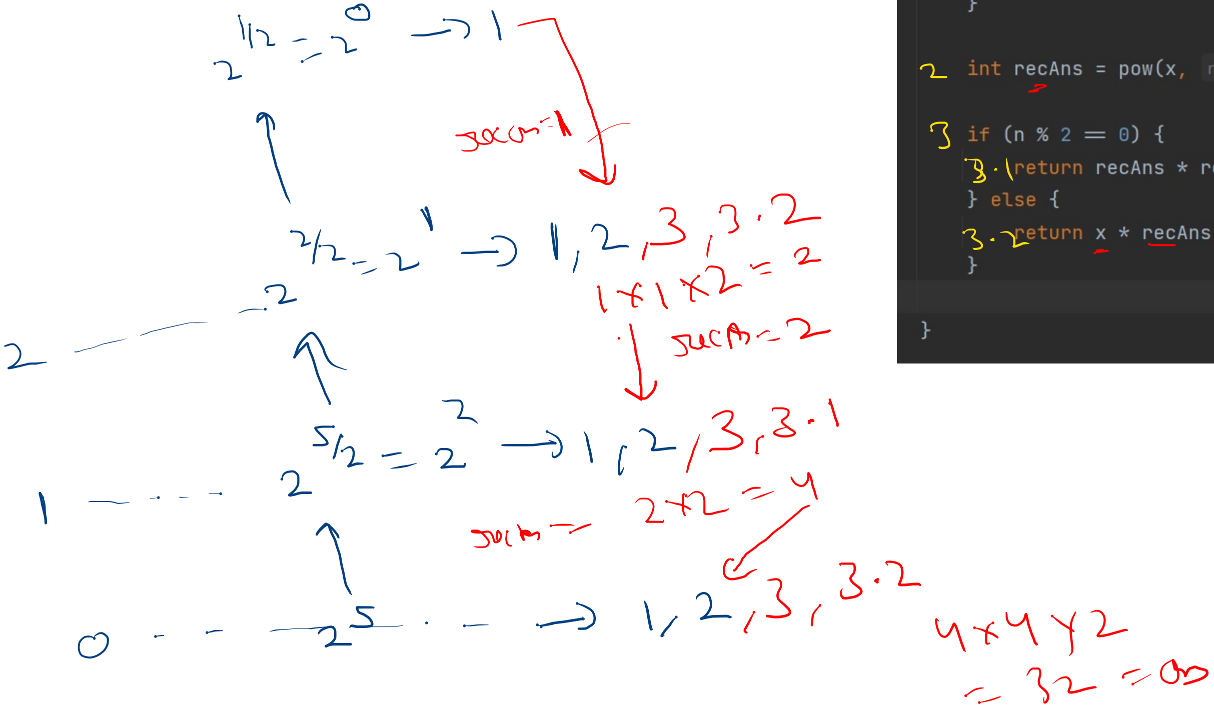
$n =$ even $\quad x^{n/2}$

$\Downarrow$

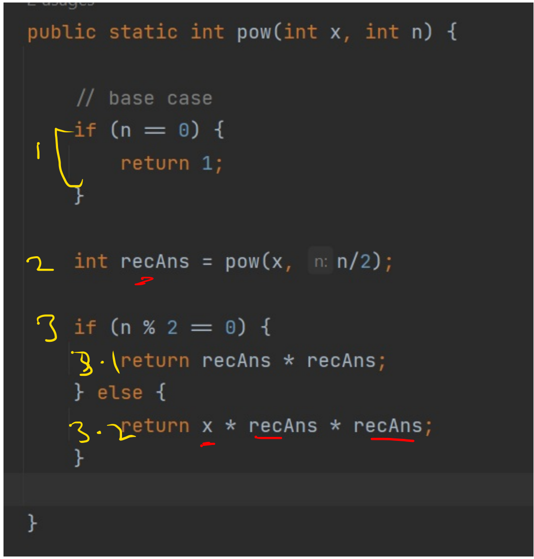
$y \times y$



$T(n) = \log_2 n$



$2^{1/2} = 2^0 \rightarrow 1$
sucm = 1
$2^{2/2} = 2^1 \rightarrow 1, 2, 3, 3.2$
1 × 1 × 2 = 2
sucAn = 2
2 ——— 2
$2^{5/2} = 2^2 \rightarrow 1, 2, 3, 3.1$
1 ---- 2
sucm = 2 × 2 = 4
$0 ---- 2^5 \rightarrow 1, 2, 3, 3.2$
4 × 4 × 2 = 32 = ans



```
public static int pow(int x, int n) {

    // base case
    if (n == 0) {
        return 1;
    }

    int recAns = pow(x, n: n/2);

    if (n % 2 == 0) {
        return recAns * recAns;
    } else {
        return x * recAns * recAns;
    }

}
```

Get Subsequence (Day 24)

Subsequence ↓ Substring

→ ⓐ b c ⓓ e ⓕ

a d f = subseq ✓

d a f = subseq ✗

a b c d e f

1). a b c → ✓

2). a b c d e f → ✓

3). b e → ✓

4). f → ✓

5). c f → ✓

6). d a b → X

b c a b d e f

11. b e f → ✓

27. b a d → ✓

37. b c d → ✓

47. a c d ✗

a b c

null ≠ empty string

a _ _

a b _

a _ c

a b c

_ _ c

_ b _

_ b c

_ _ _

8 Subseq

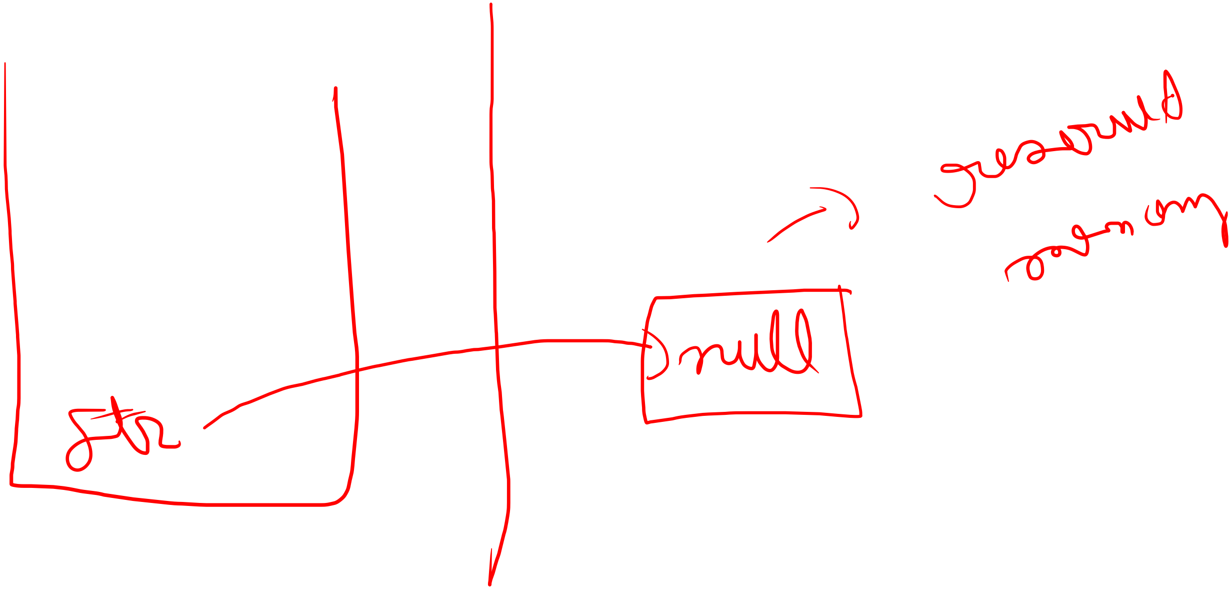
n = len of str

$\boxed{\text{Total Subseq} = 2^n}$

$2^3 = 8$

null → special keyword

= associated with non primitive datatypes



a b c

a _ _

a b _

a _ c

a b c

_ _ c

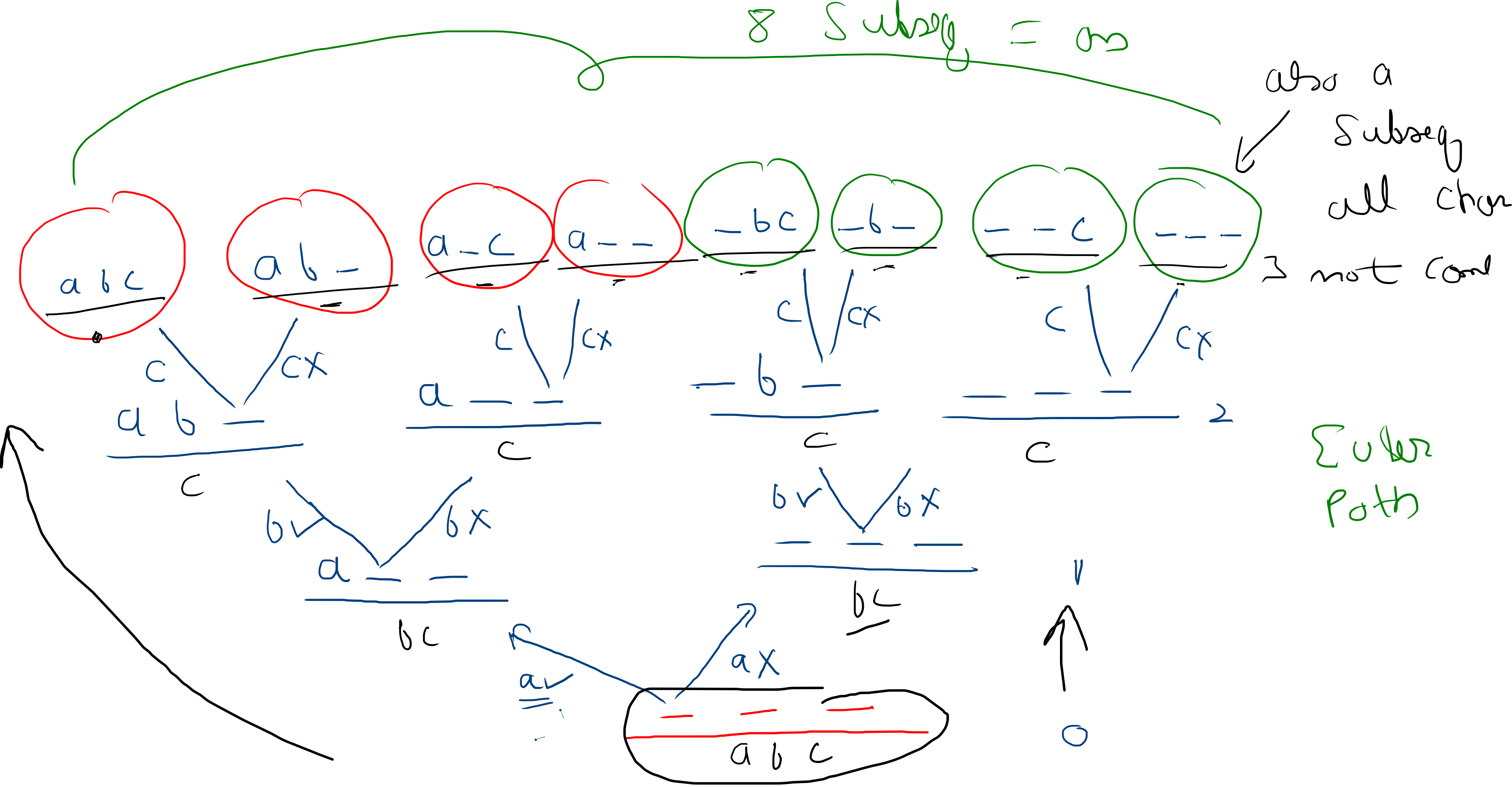
_ b _

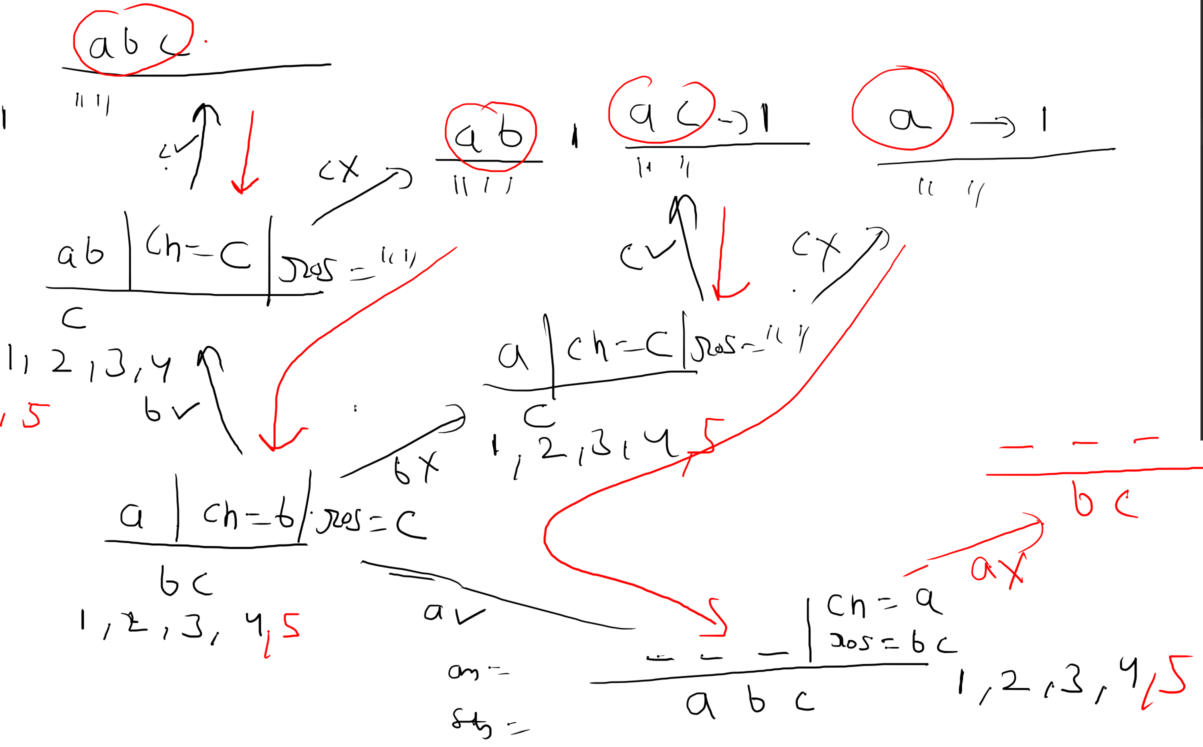
_ b c

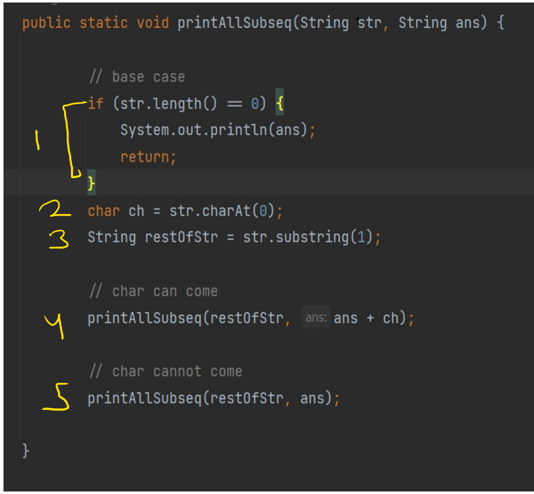
_ _ _

for a single char Position, a char can come or not come

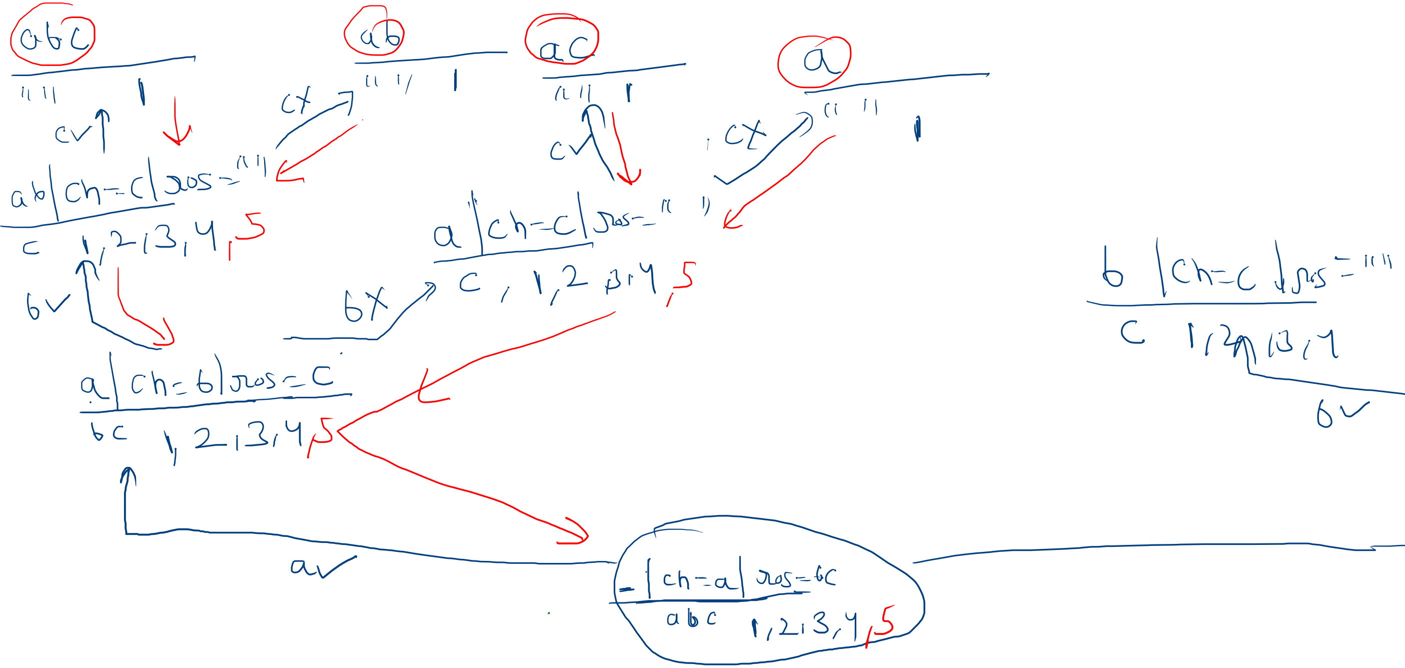
_ b c

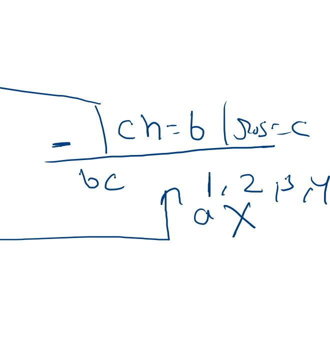

8 Subseq = ans
also a Subseq
all char
∃ not cond
abc
ab_
a_c
a__
_bc
_b_
__c
___
c
cx
ab_
c
a__
c
_b_
c
___
c
2
bv
bx
a__
bc
bv
bx
_ _ _
bc
1
av
ax
_ _ _
abc
0
Euler Path

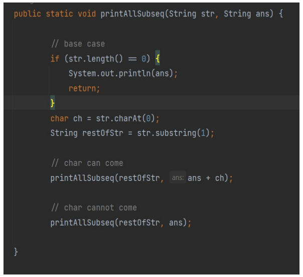




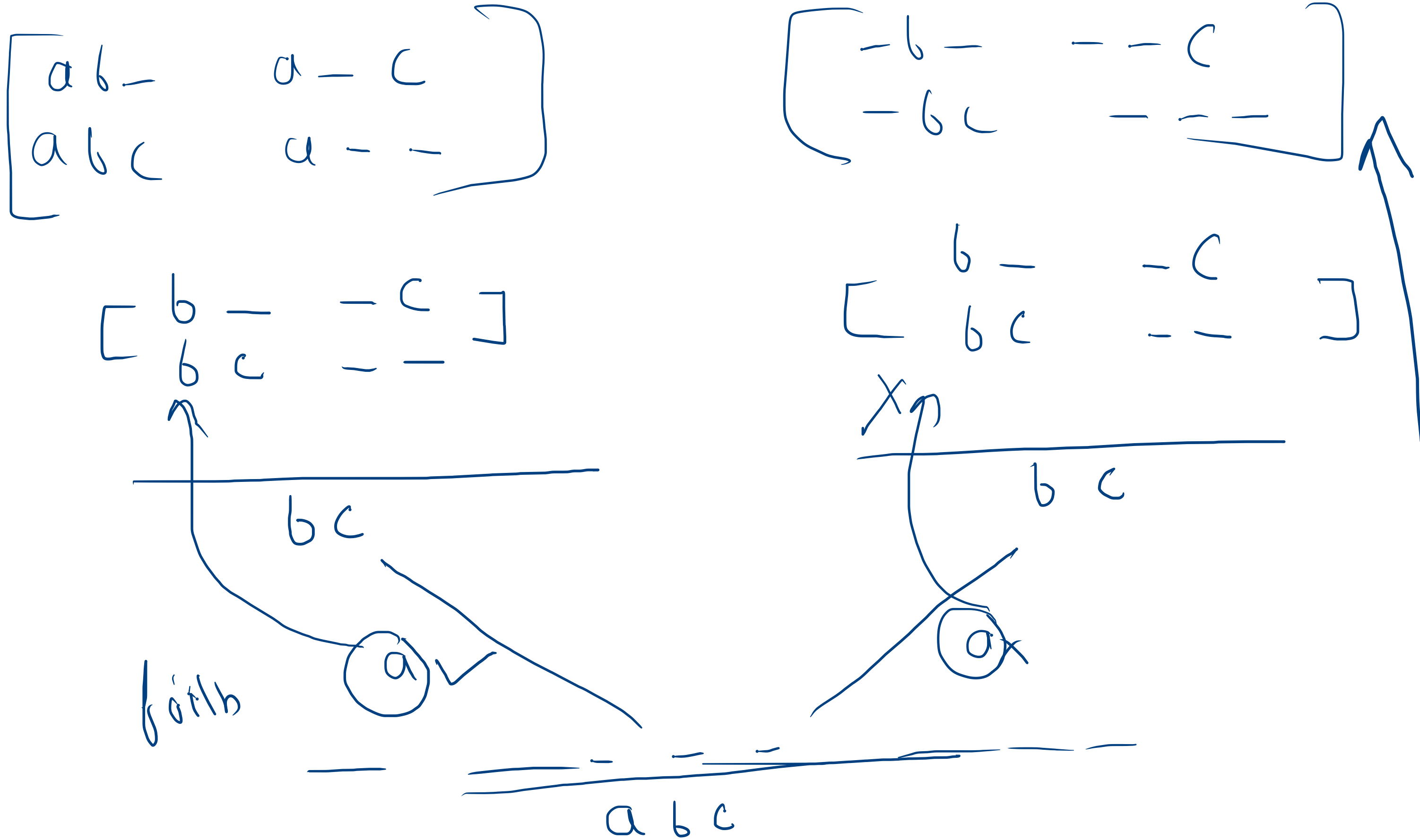
```java
public static void printAllSubseq(String str, String ans) {

    // base case
    if (str.length() == 0) {
        System.out.println(ans);
        return;
    }
    char ch = str.charAt(0);
    String restOfStr = str.substring(1);

    // char can come
    printAllSubseq(restOfStr, ans: ans + ch);

    // char cannot come
    printAllSubseq(restOfStr, ans);

}
```







```java
public static void printAllSubseq(String str, String ans) {

        // base case
        if (str.length() == 0) {
            System.out.println(ans);
            return;
        }
        char ch = str.charAt(0);
        String restOfStr = str.substring(1);

        // char can come
        printAllSubseq(restOfStr, ans: ans + ch);

        // char cannot come
        printAllSubseq(restOfStr, ans);

}
```



ab_ a_c
abc a__
_b_ __c
_bc ___
b_ _c
bc __
b_ _c
bc __
x
bc
bc
a ✓
a x
faith
abc