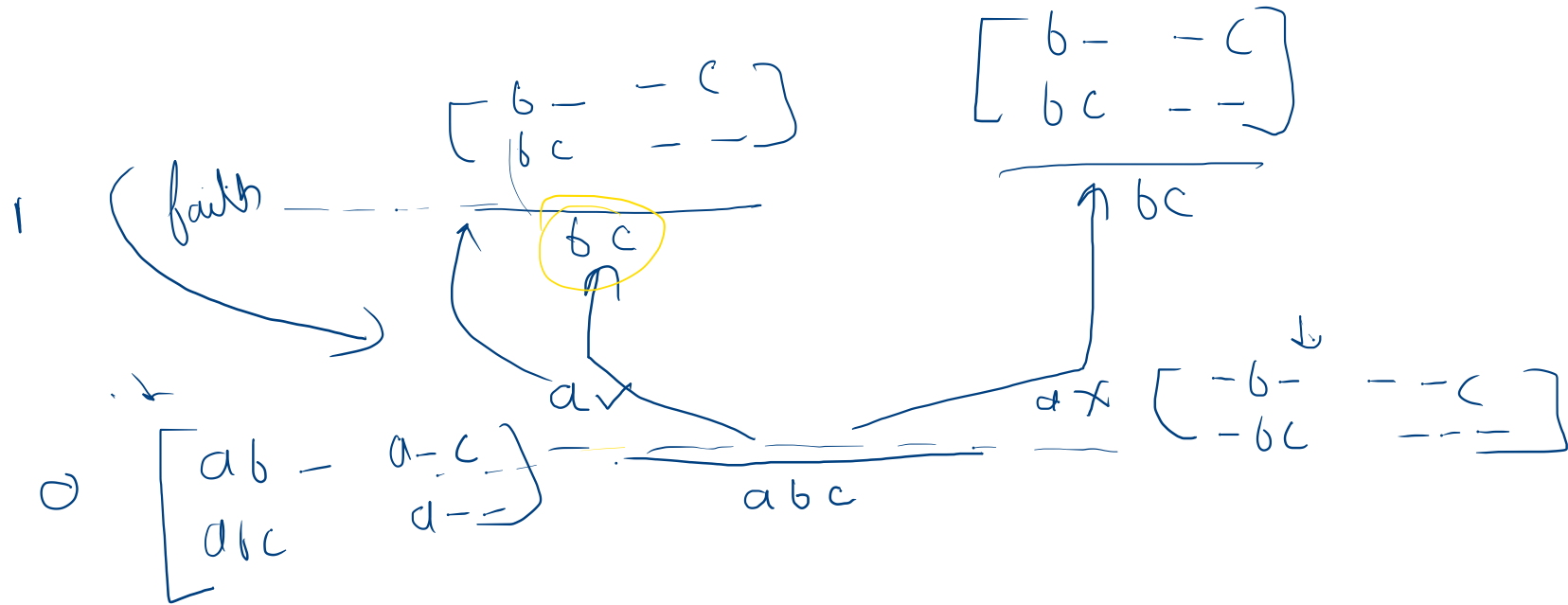


Get Subsequence (Day 24)

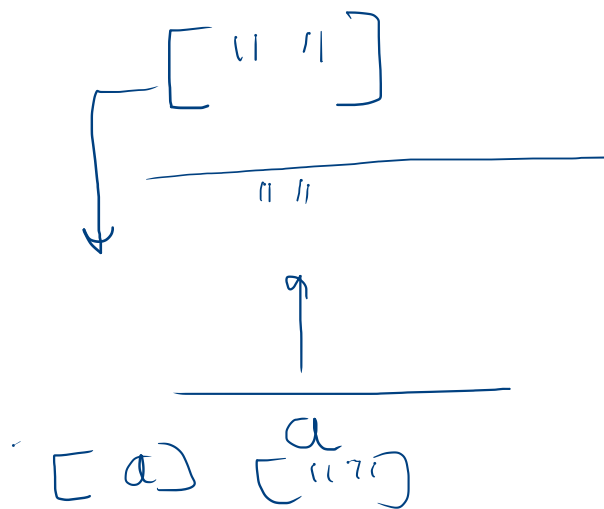


base case

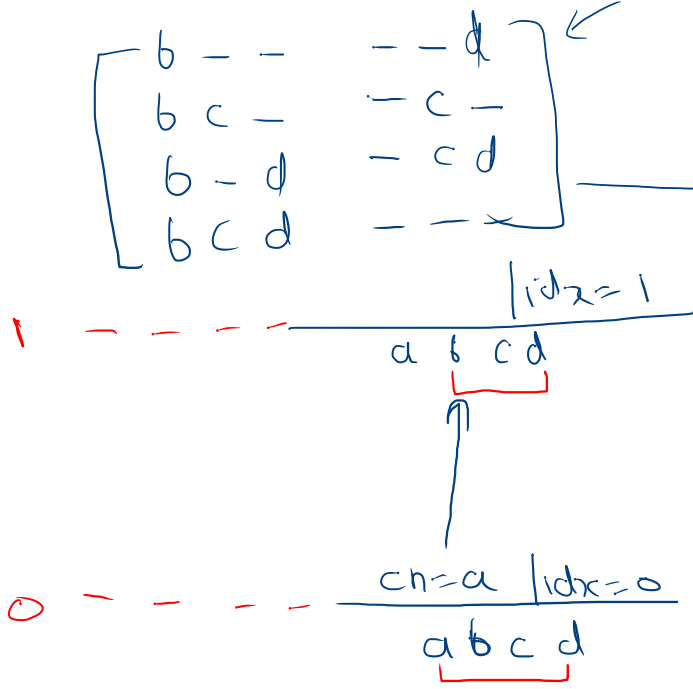
$\rightarrow (n-1)$ level $\rightarrow n^{\text{th}}$ level

$[] \rightarrow$ empty AL
no element

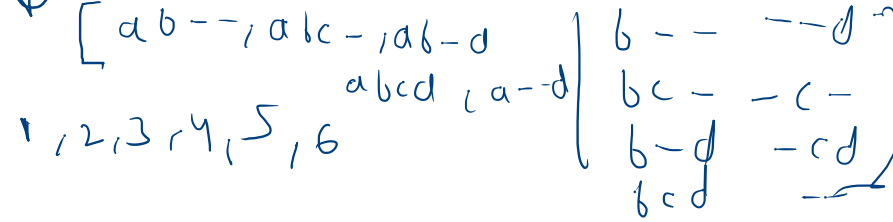
$[""] \rightarrow$ one element
element = empty str



(both)



my ans



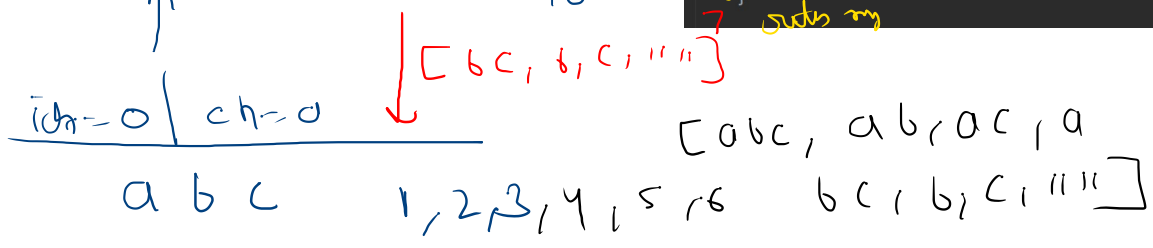
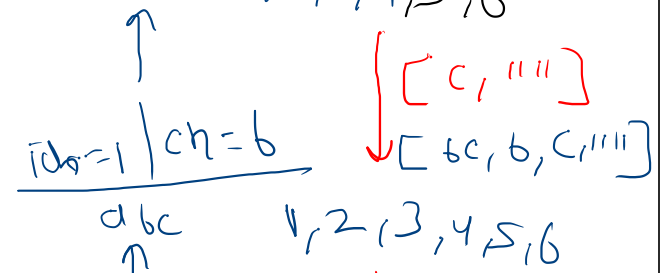
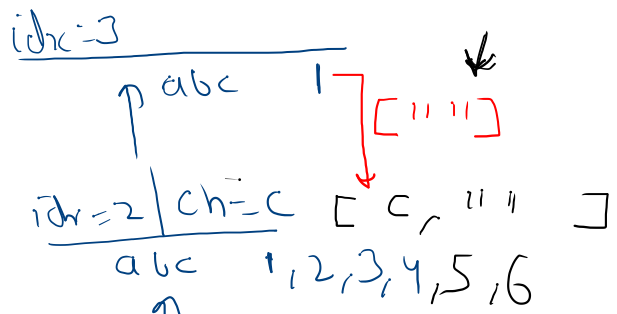
```
if (idx == str.length()) {  
    ArrayList<String> bans = new ArrayList<>();  
    bans.add("");  
    return bans;  
}  
  
2 char ch = str.charAt(idx);  
  
    // faith call  
3 ArrayList<String> recAns = printAllSubseq4Best(str, idx + 1);  
  
4 ArrayList<String> myAns = new ArrayList<>();  
  
    // char will come  
5 for (String val: recAns) {  
        myAns.add(ch + val);  
    }  
  
    // char will not come  
6 for (String val: recAns) {  
        myAns.add(val);  
    }  
}
```

3

2

1

0



```
if (idx == str.length()) {  
    ArrayList<String> bans = new ArrayList<>();  
    bans.add("");  
    return bans;  
}  
  
char ch = str.charAt(idx);  
  
// faith call  
ArrayList<String> recAns = printAllSubseq4Best(str, idx + 1);  
  
ArrayList<String> myAns = new ArrayList<>();  
  
// char will come  
for (String val: recAns) {  
    myAns.add(ch + val);  
}  
  
// char will not come  
for (String val: recAns) {  
    myAns.add(val);  
}
```

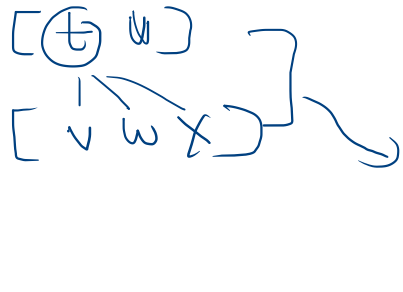
[abc, ab, ac, a, bc, b, c, ""]

Get Kpc (Day 25)

Nokia

	<u>0</u>	
<u>1</u>	<u>2</u>	<u>3</u>
abc	def	ghi
<u>4</u>	<u>5</u>	<u>6</u>
jkl	mno	pqr
<u>7</u>	<u>8</u>	<u>9</u>
tu	vwx	yz

7 8



tu ex uv wx
 tu ex uv wx

Code

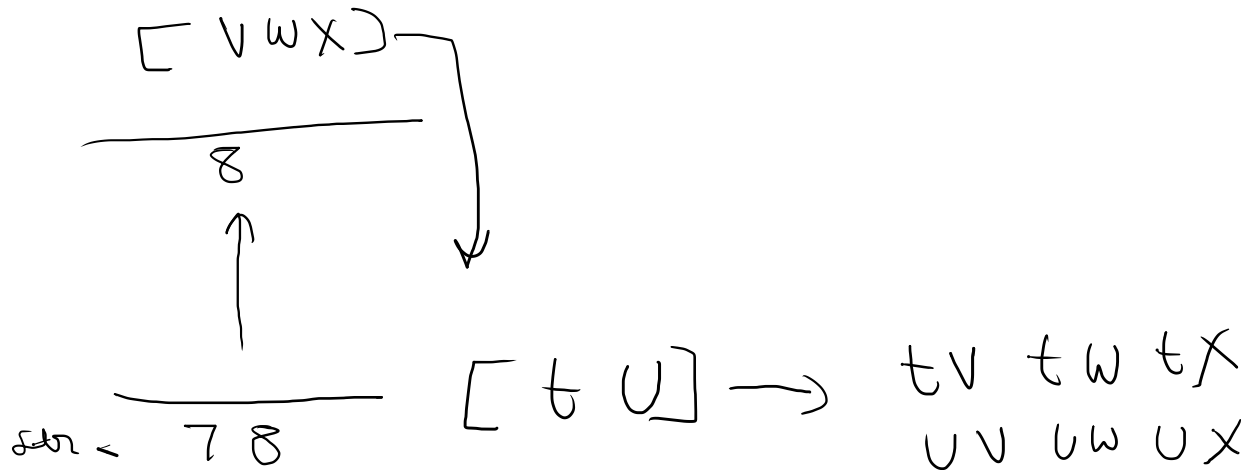
1 5 2 2
 C O d e

1 2 3

a d g a e g a b a
 h h h
 i i i

```
String[] codes = {".", "abc", "def", "ghi", "jkl", "mno", "pqrs", "tu", "vwx", "yz"};
```

final



fault

[tv tw tx
uv uw ux]

78

↑
178

[a b c]

a	b tv	b tw	b tx
b	b uv	b uw	b ux
c	c tv	c tw	c tx
	c uv	c uw	c ux

atv atw atr
a uv a uw a ux

0 1 2 3 4 5 6 7 8 9

```
static String[] codes = {".", "abc", "def", "ghi", "jkl", "mno", "pqrs", "tu", "vwx", "yz"};
```

'0' — 48

(1) → 49

(2) → 50

ch = '2' → 50

ch - '0'

50 - 48 = 2

```
public static List<String> solution(String str, int idx) {  
  
    // base case  
    if (idx == str.length()) {  
        List<String> bans = new ArrayList<>();  
        bans.add("");  
        return bans;  
    }  
  
    char ch = str.charAt(idx); // 7  
  
    List<String> recAns = solution(str, idx + 1);  
  
    String currCode = codes[ch - '0'];  
  
    List<String> myAns = new ArrayList<>();  
    for (char cch: currCode.toCharArray()) {  
        for (String recStr: recAns) {  
            myAns.add(cch + recStr);  
        }  
    }  
  
    return myAns;  
}
```


0 1 2 3 4 5 6 7 8 9

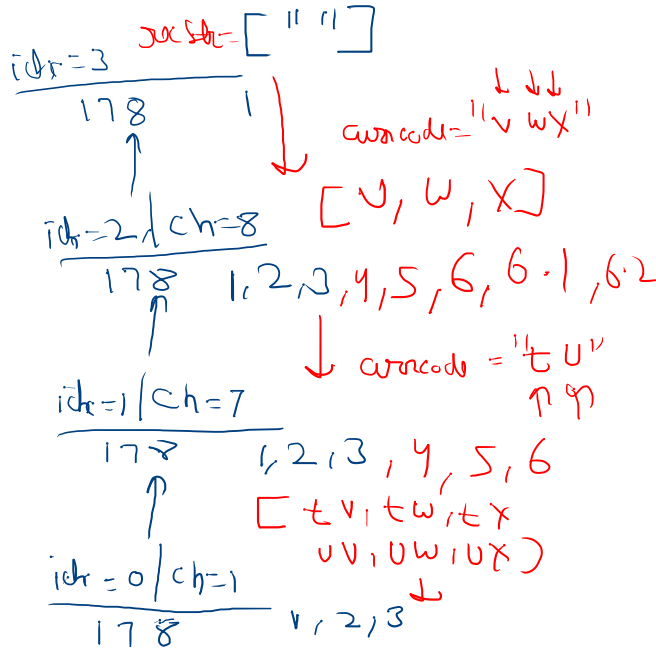
static String[] codes = {".", "abc", "def", "ghi", "jkl", "mno", "pqrs", "tu", "vw", "yz"};

currCode = "tU"
↑↑

[v, w, x]

t → v, w, x

U → v, w, x



```
public static List<String> solution(String str, int idx) {

    // base Case
    if (idx == str.length()) {
        List<String> bans = new ArrayList<>();
        bans.add("");
        return bans;
    }

    2 char ch = str.charAt(idx); // 7

    3 List<String> recAns = solution(str, idx + 1);

    4 String currCode = codes[ch - '0'];

    5 List<String> myAns = new ArrayList<>();
    6 for (char cch: currCode.toCharArray()) {
        6.1 for (String recStr: recAns) {
            6.2 myAns.add(cch + recStr);
        }
    }

    7 return myAns;
}
```