

→ Recursion on the way up.

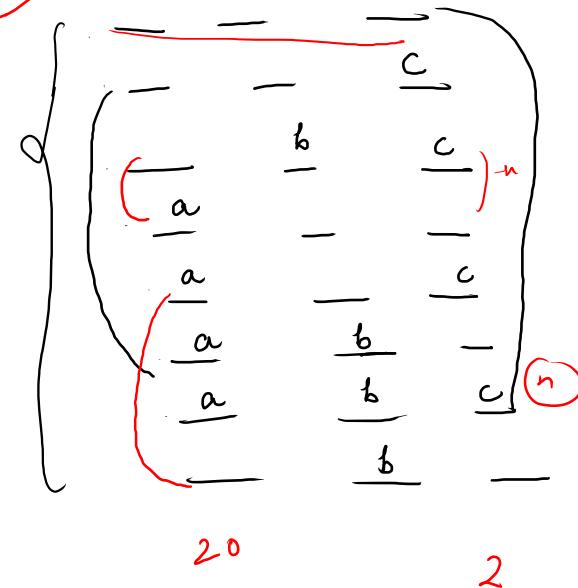
→ Get subsequence x x Recursion with array list

str → "abc"
 \|/
 2ⁿ * 2ⁿ

8

→ n → 2ⁿ

char → 2 bytes



⇒ 2ⁿ * n * 2
bytes ↓

→ 2ⁿ * n

2²⁰ * 20
bytes.

(Recursion on the way up) / (Question / answer approach)

get subsequence

$$\Rightarrow 2^n \times n$$

Recurⁿ
with
array list



Every character has
2 choices

- Include
- Exclude

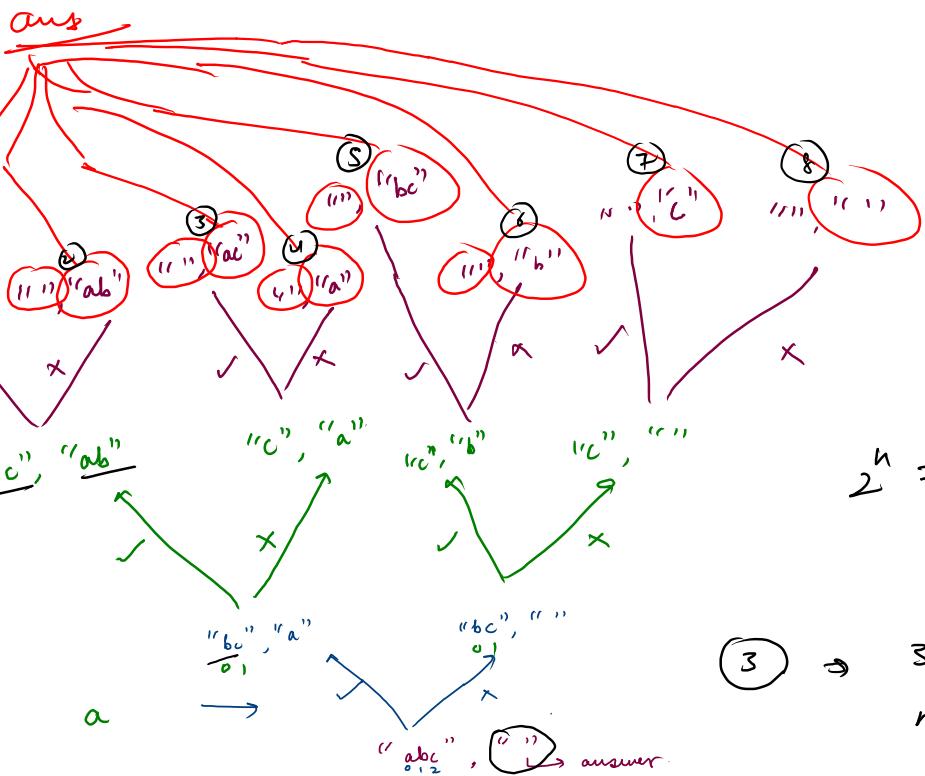
if $\text{len}(\text{ans}) = n$

c →

b →



a



$$2^n \times n$$

$$2^n = 2^3 = 8$$

(3)

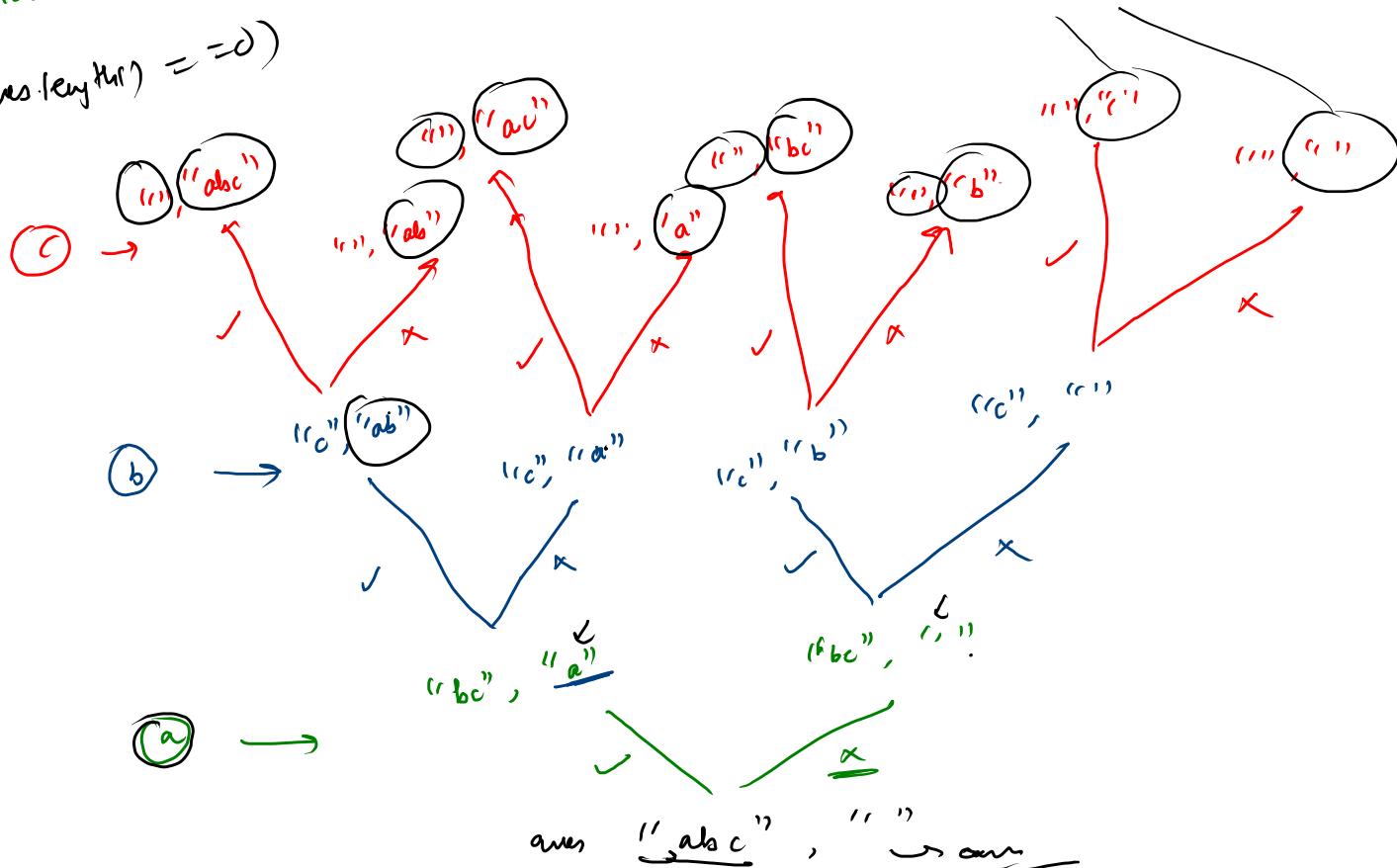
$$3 \times 2 = 6$$

$$n \times 2 = \underline{2n \text{ lengths}}$$

answer.

↳ Include
↳ Exclude

if(ques(key[k]) == -1)



⇒ extend 1 hr.



11:10 pm



```

import java.io.*;
import java.util.*;

public class Solution {
    abc    //①
    public static void printSS(String ques, String ans){
        if(ques.length() == 0){
            System.out.println(ans);
            return;
        }

        char ch = ques.charAt(0);
        String ros = ques.substring(1)); -②
        // Include
        printSS(ros, ans + ch); -③
        // Exclude
        printSS(ros,ans); -④
    }

    public static void main(String[] args) {
        // Enter your code here. Read input from STDIN. Print output to STDOUT. Your
        Scanner scn = new Scanner(System.in);
        String ques = scn.next();
        printSS(ques, "");
    }
}

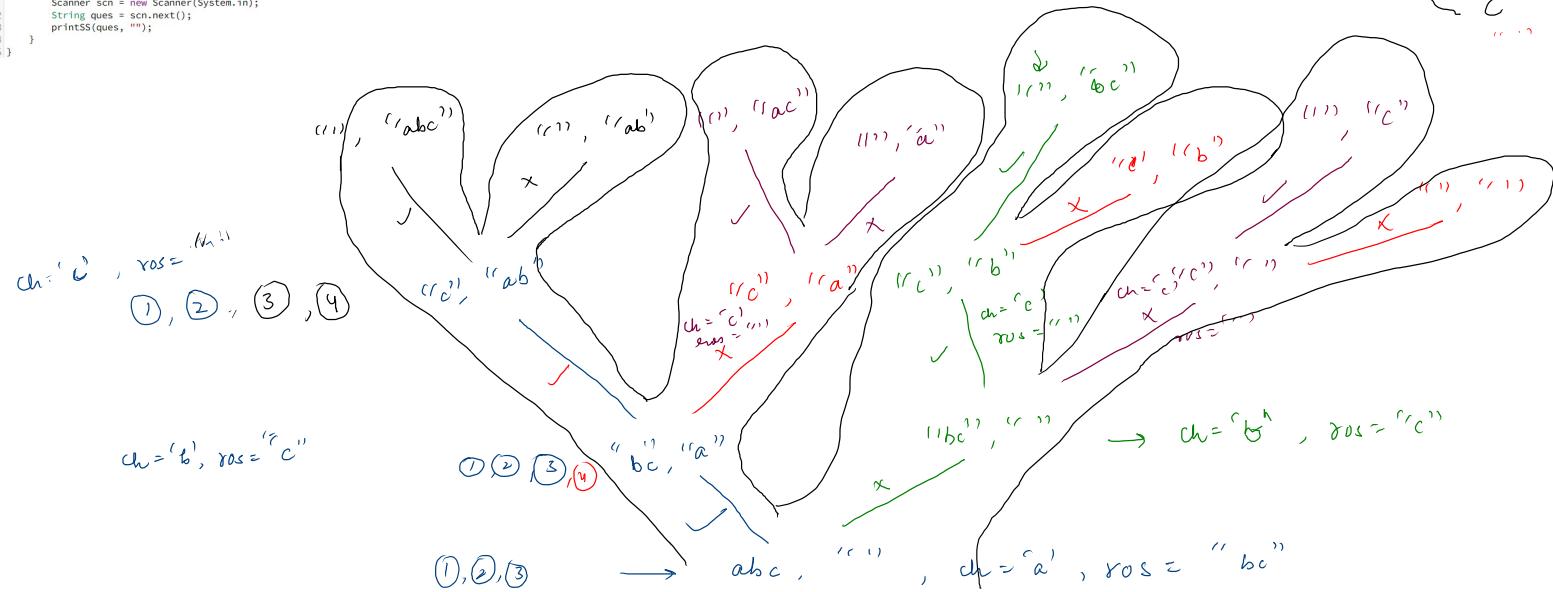
```

Code + Dry Run → 88 SS

≡ +
print KPE

Dry Run

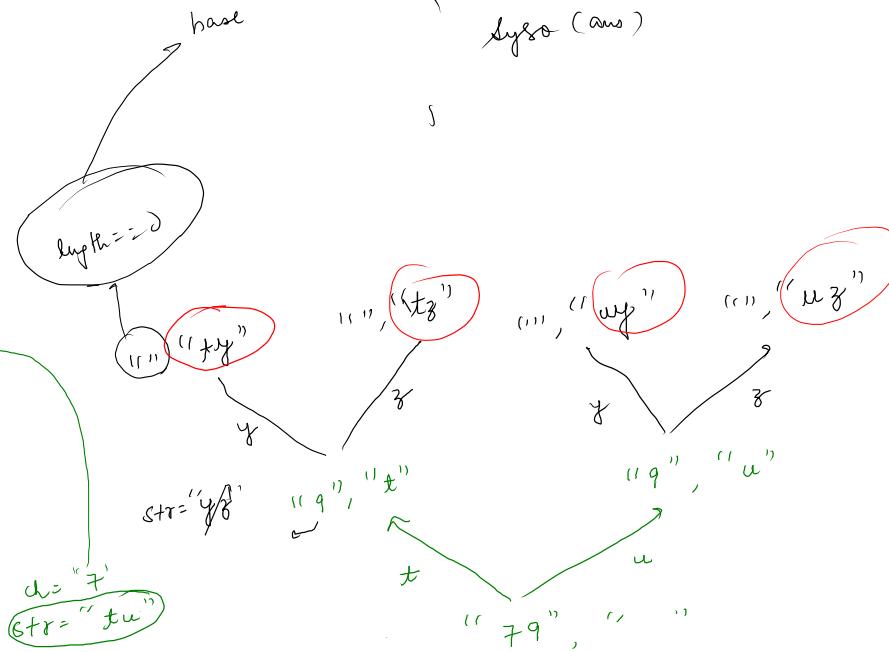
abc
ab
ac
a
bc
b
c



Paint KPC

"79"

$0 \rightarrow "i;"$
 $1 \rightarrow "abc"$
 $2 \rightarrow "def"$
 $3 \rightarrow "ghi"$
 $4 \rightarrow "jkl"$
 $5 \rightarrow "mno"$
 $6 \rightarrow "pqrs"$
 $7 \rightarrow "tuv"$
 $8 \rightarrow "vwx"$
 $9 \rightarrow "y়z"$



10 mins →

Code + Day run

+ Print permutations

```
public class Solution {  
    public static String [] arr = {".;","abc","def","ghi","jkl","mno","pqrs","tu","vwx","yz"};  
    public static void printKPC(String ques, String ans){  
        if(ques.length() == 0){  
            System.out.println(ans);  
            return;  
        }  
  
        char ch = ques.charAt(0);  
        String ros = ques.substring(1);  
  
        String str = arr[ch-'0'];  
        for(int i=0;i<str.length();i++){  
            char c = str.charAt(i);  
            printKPC(ros,ans + c);  
        }  
    }  
  
    public static void main(String[] args) {  
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named  
        Scanner scn = new Scanner(System.in);  
        // String ques = scn.next();  
        // printKPC(ques,"");  
        printKPC(scn.next(), "");  
    }  
}
```

Print Permutations (Day 26)

Sample Input 0

Problem

Submissions

Leaderboard

Discussions

abc

1. You are given a string str.

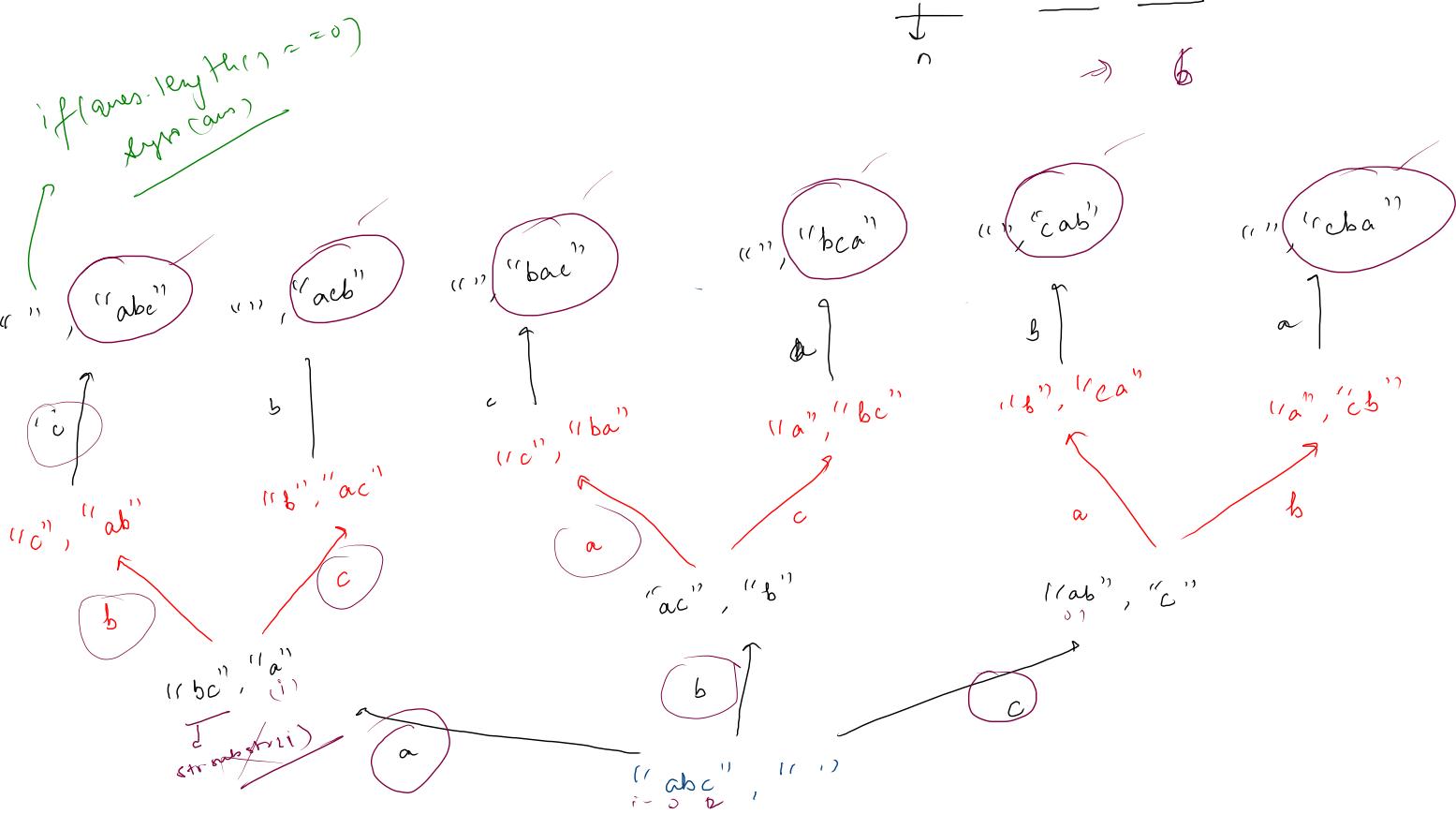
2. Calculate and print all permutations of str. Use sample input and output to take idea about permutations.

$$str = "abc"$$

Sample Output 0

abc
acb
bac
bca
cab
cba

$$\begin{array}{r} 3 \\ \times 2 \\ \hline a/b/c \end{array} \quad \times \quad \begin{array}{r} 2 \\ \times 1 \\ \hline b/c \\ a/c \\ a/b \end{array} \quad = \quad 3 \times 2 \times 1 = 6$$



$\text{tos} = \text{ans} \cdot \text{subString}^{(1)}$

Code + Dry Run

⇒ 10 min

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5     public static void printPermutations(String ques, String ans){
6         if(ques.length() == 0){
7             System.out.println(ans);
8             return;
9         }
10
11        for(int i=0;i<ques.length();i++){
12            char ch = ques.charAt(i);
13            String ros = ques.substring(0,i) + ques.substring(i+1);
14            printPermutations(ros,ans + ch);
15        }
16    }
17
18    public static void main(String[] args) {
19        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should
20        Scanner scn = new Scanner(System.in);
21        printPermutations(scn.next(), "");
22    }
23 }
```

1. You are given a string str of digits. (will never start with a 0)

2. You are required to encode the str as per following rules 1 -> a 2 -> b 3 -> c .. 25 -> y 26 -> z

3. Calculate and print all encodings of str. Use the input-output below to get more understanding on what is required
123 -> abc, aw, lc 993 -> 013 -> Invalid input. A string starting with 0 will not be passed. 103 -> jc
303 -> No output possible. But such a string maybe passed. In this case print nothing.

$str = "655198"$

~~993~~

123

abc
aw
aw

x
123
w
abc
aw
aw

993 -> i30
c-24

?

993 -> i30
c-24
abc
aw
aw

103 -> jc
10 3
a
10 3

$\begin{cases} 1 \rightarrow a \\ 2 \rightarrow b \\ 3 \rightarrow c \\ 4 \rightarrow d \\ 5 \rightarrow e \\ 6 \rightarrow f \\ 7 \rightarrow g \\ 8 \rightarrow h \\ 9 \rightarrow i \\ 10 \rightarrow j \\ 11 \rightarrow k \\ 12 \rightarrow l \\ 13 \rightarrow m \\ 14 \rightarrow n \\ 15 \rightarrow o \\ 16 \rightarrow p \\ 17 \rightarrow q \\ 18 \rightarrow r \\ 19 \rightarrow s \\ 20 \rightarrow t \\ 21 \rightarrow u \\ 22 \rightarrow v \\ 23 \rightarrow w \\ 24 \rightarrow x \\ 25 \rightarrow y \\ 26 \rightarrow z \end{cases}$

Sample Input 0

655196

Sample Output 0

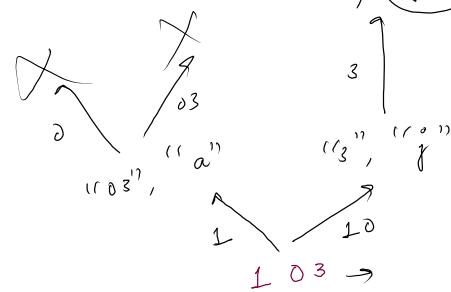
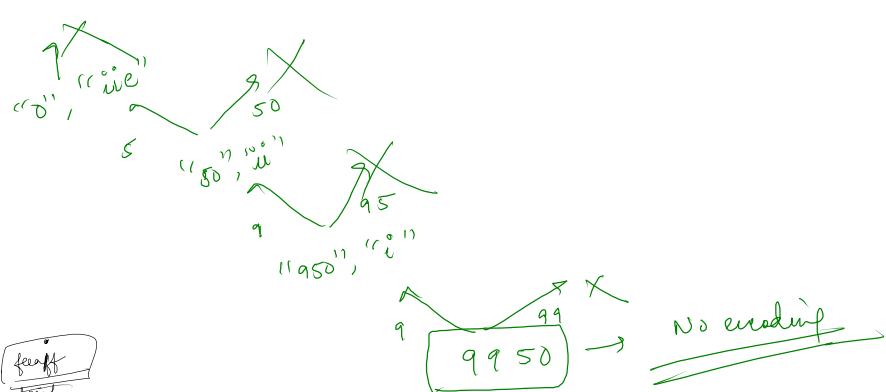
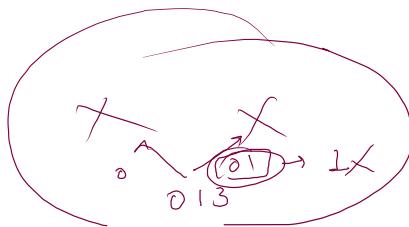
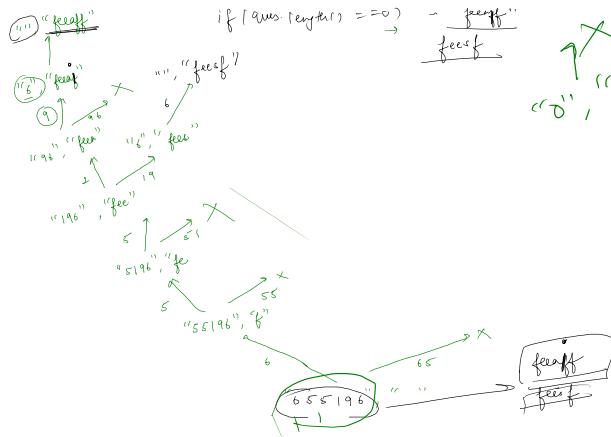
feeaif
feesf

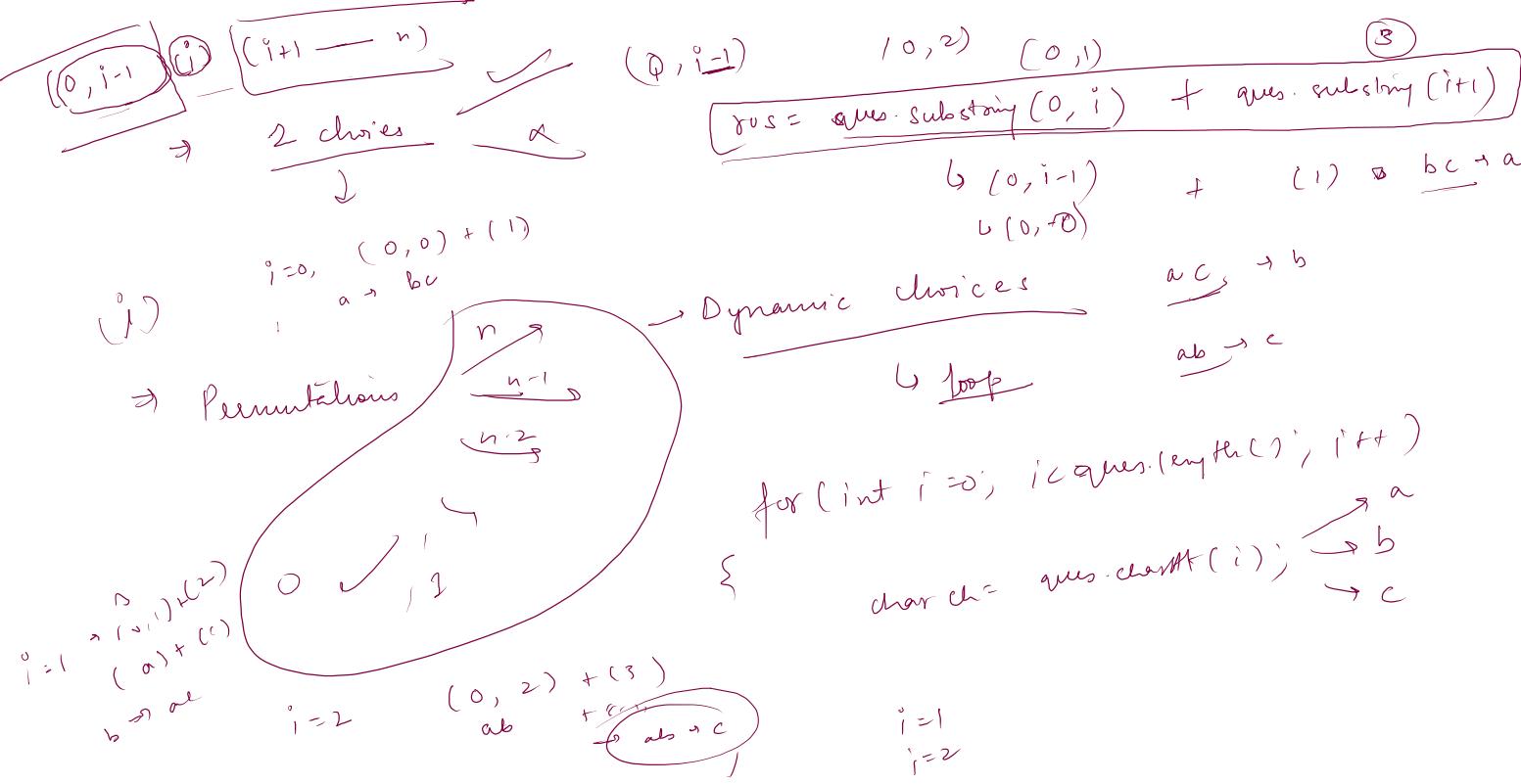
10
20
30
40
50

$\rightarrow '0'$ as a single digit \rightarrow no answer possible

$\rightarrow '10'$ as double digit \rightarrow answer may or not possible

1 → a
 2 → b
 3 → c
 4 → d
 5 → e
 6 → f
 7 → g
 8 → h
 9 → i
 0 → j
 1 → k
 2 → l
 ; → m
 , → n
 5 → o
 6 → p
 7 → q
 8 → r
 9 → s
 - → t
 , → u
 2 → v
 3 → w
 4 → x
 5 → y
 2 → z





strong \Rightarrow "abcde^f"
 0 1 2 3 4 5

excluded
 substring(1, 4) \Rightarrow bcd
 substring(1, 5) \Rightarrow bcd

$$\begin{array}{r}
 97 \\
 6 \\
 \hline
 103
 \end{array}$$

655

$6 \rightarrow f$

(char) $\boxed{6 + 'a' - 1}$

$$6 + \cdot \Rightarrow + - 1$$

97	$\rightarrow a$
98	$\rightarrow b$
99	$\rightarrow c$
100	$\rightarrow d$
101	$\rightarrow e$
102	$\rightarrow f$
103	$\rightarrow g$

decoding $\Rightarrow (\text{char})(\text{ch} + 'a' - 1)$

```
String decoding2 = Integer.parseInt(ques.substring(0,2))
```



Integer.parseInt("65")

(65) → ~~Integer~~

A diagram illustrating the substring `ques.substring(0,2)`. It shows the string "65" with a circled '6' at index 0 and a circled '5' at index 1. Below the string, the indices 0, 1, and 2 are written.

```

public static void printEncodings(String ques, String ans){
    if(ques.length() == 0) {
        System.out.println(ans);
        return;
    }
    char ch = ques.charAt(0);
    if(ch == '0'){
        return;
    }
    int num1 = ch-'0';
    char decoding1 = (char)(num1-1+'a');
    String ros1 = ques.substring(1);
    printEncodings(ros1, ans + decoding1);

    if(ques.length() >= 2){
        int num = Integer.parseInt(ques.substring(0,2));
        if(num <=26){
            char decoding2 = (char)(num+ 'a'-1);
            // System.out.println(num);
            String ros2 = ques.substring(2);
            printEncodings(ros2, ans + decoding2);
        }
    }
}

```

$$\begin{aligned}
 \text{ch} &\rightarrow '3' - 3 + 97 - 1 = 99 \rightarrow 'c' \\
 \text{num1} &= 3 \\
 \text{decoding1} &= 3 + ('x') - 1 = 34 \\
 \text{decoding1} &= (\text{char}) 3 + (97) - 1 \rightarrow 'a' \\
 \text{num1} &= 1 \\
 \text{ch} &\rightarrow '2' - 6
 \end{aligned}$$

$q \rightarrow$
 98
 $99 \rightarrow 00 \rightarrow d$

$101 \rightarrow$
 $2 \rightarrow f$
 $3 \rightarrow g$
 $4 \rightarrow h$
 $5 \rightarrow i$
 $6 \rightarrow j$
 $7 \rightarrow k$
 $8 \rightarrow l$
 $9 \rightarrow m$

~~198~~
26

ack

bad



"ack"

"v11"

"u"

"a"

"c"

"34", "a"

"34", "

34

"l"

13

134, "

"nl d"

"f"

"g"

"h"

"i"

"j"

"k"

"l"

decoding2 = "13 + ('a'-1)"
 $18 + 97$
109

