12

↳ 6 → Backlog

↳ 6 → now

↳ Current + extra leetcode

⇒

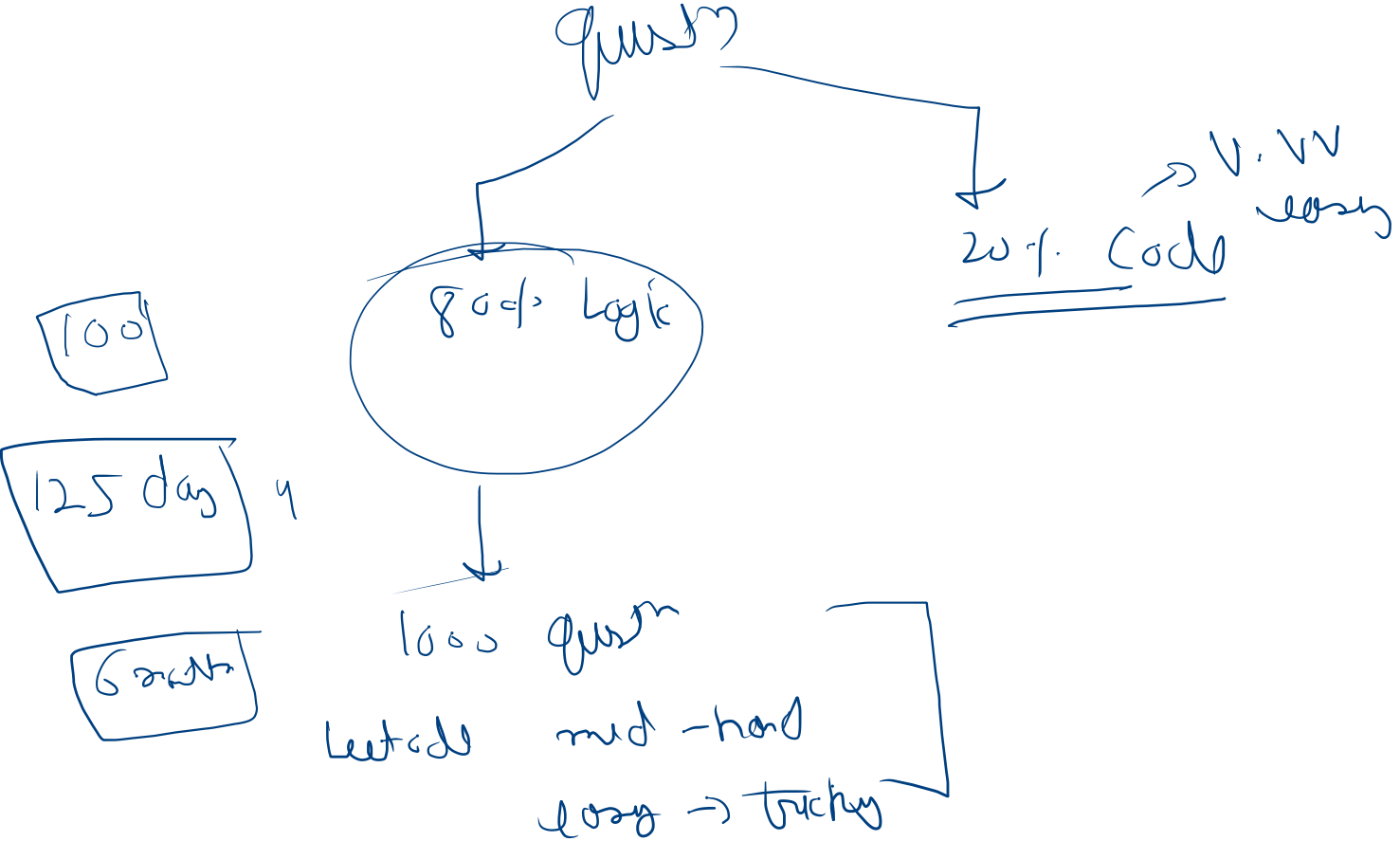
Que

now Que

→ Core Subjects

OOPS OS

PBMs CN → Books

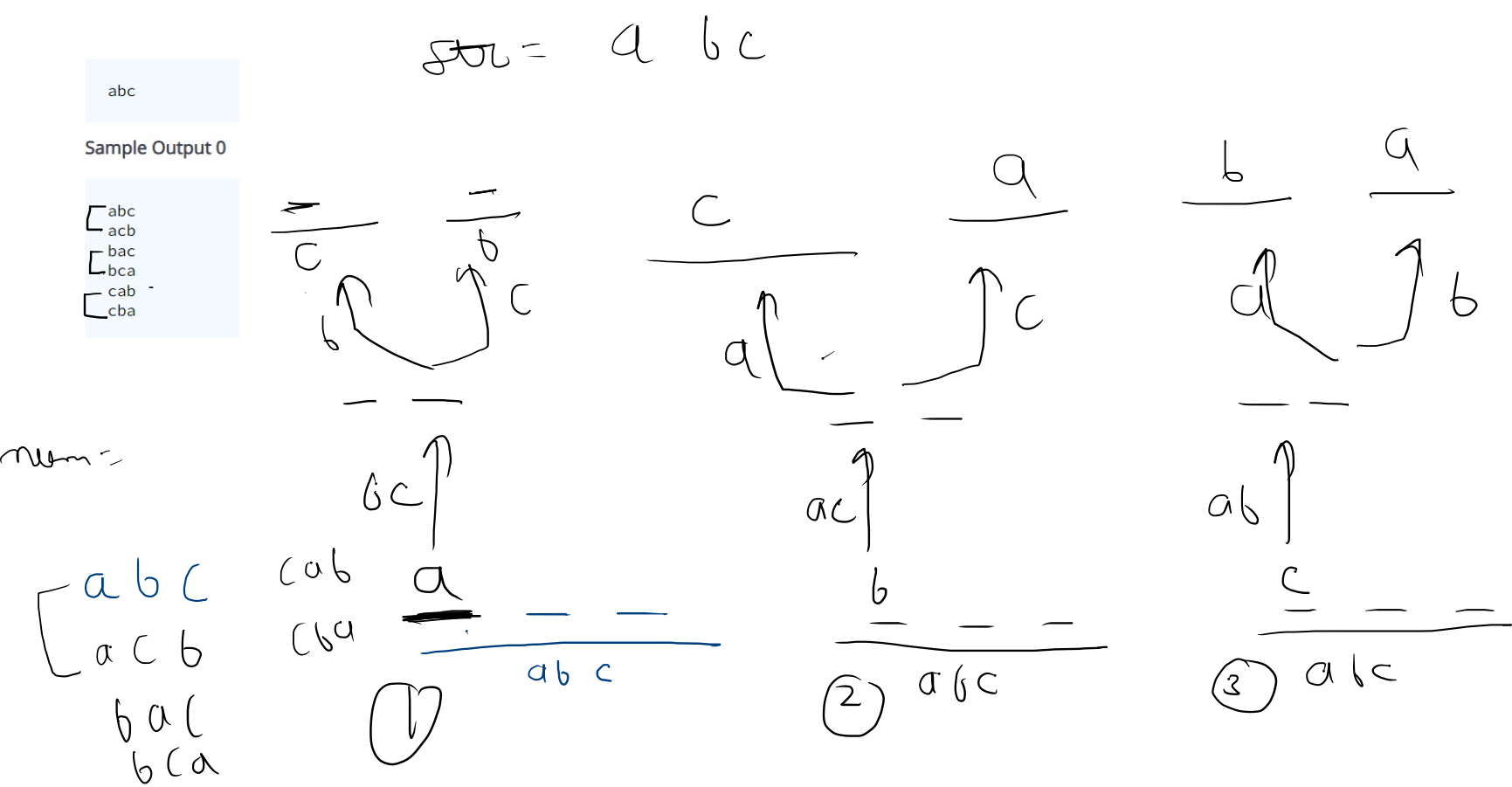

Ques
20% Code
80% Logic
100
125 days
6 months
1000 Ques
Leetcode
mid - hard
easy → tricky

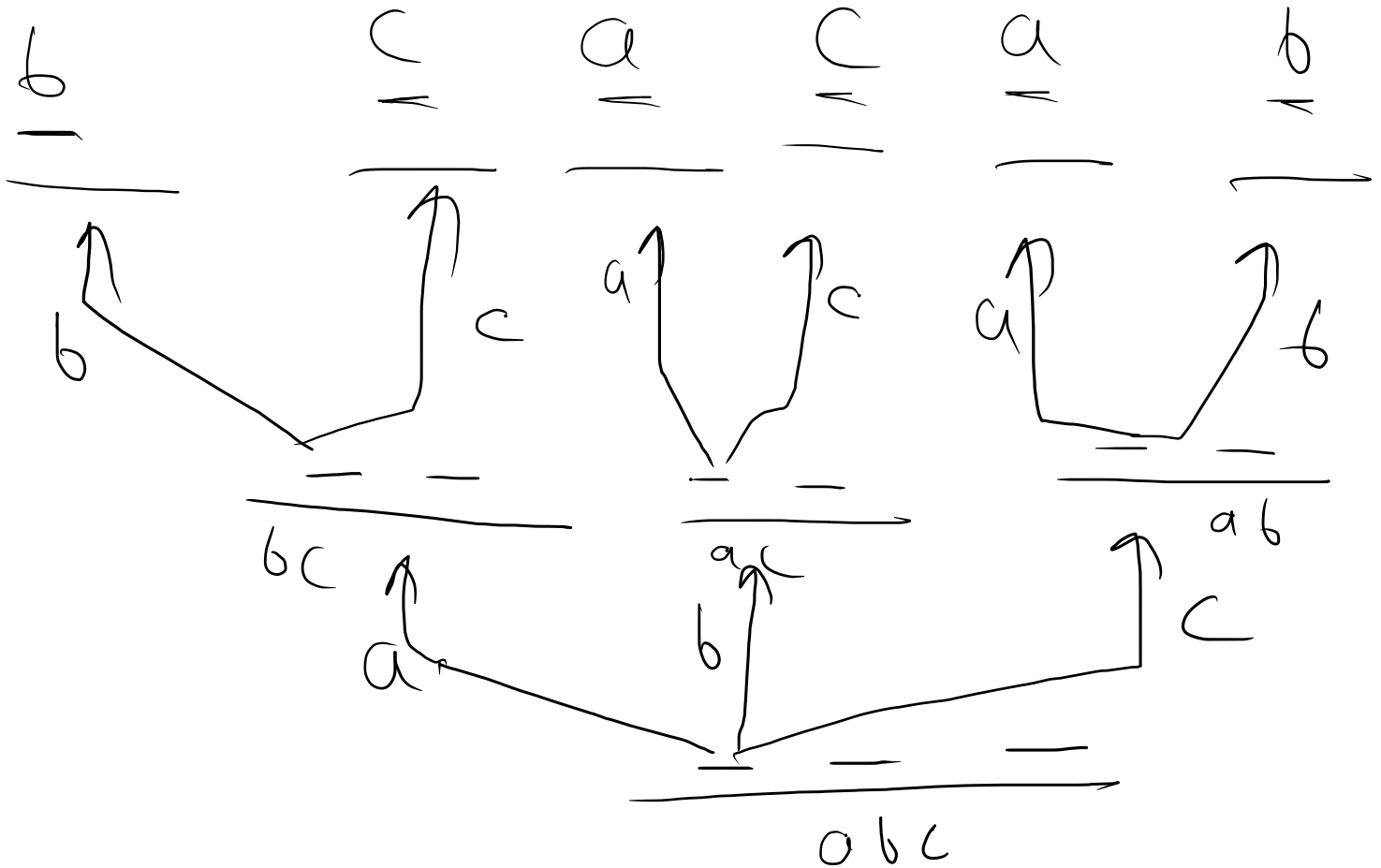
Print Permutations (Day 26)

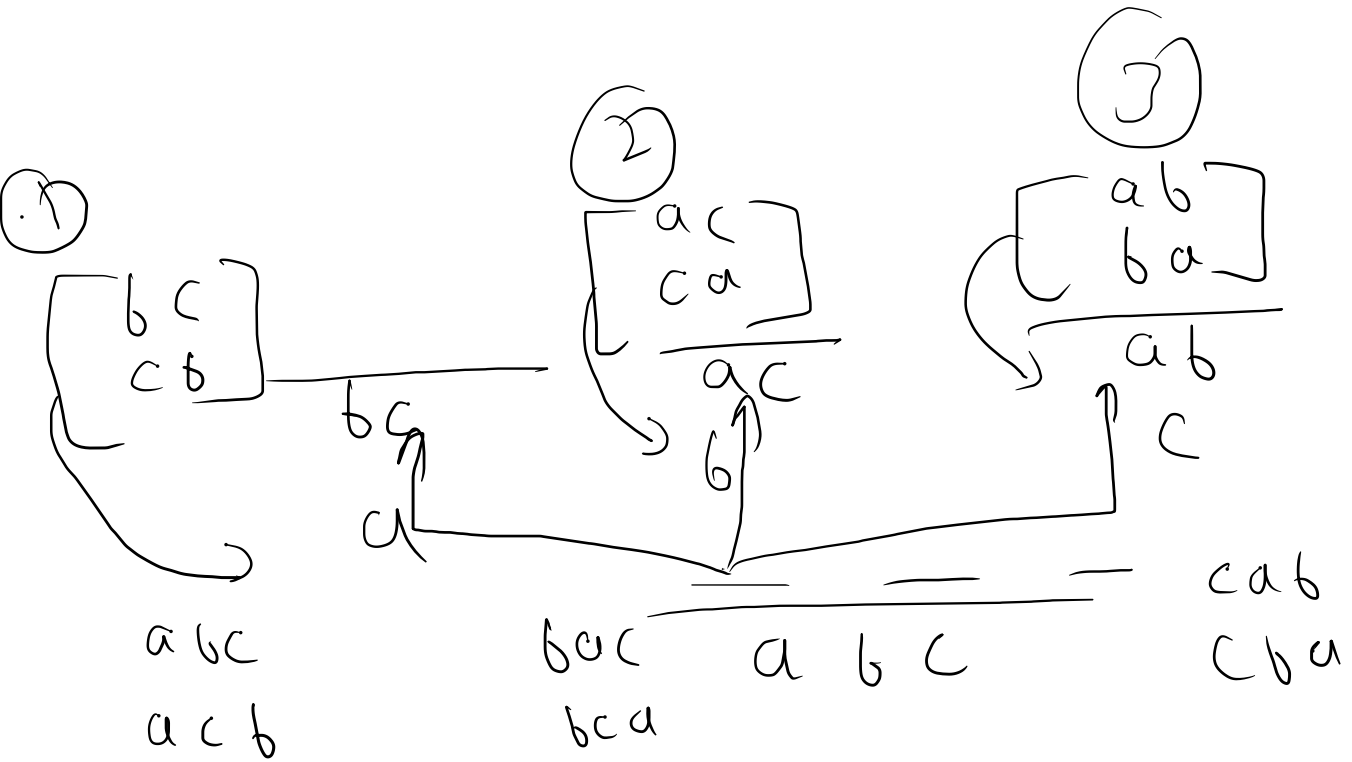
abc

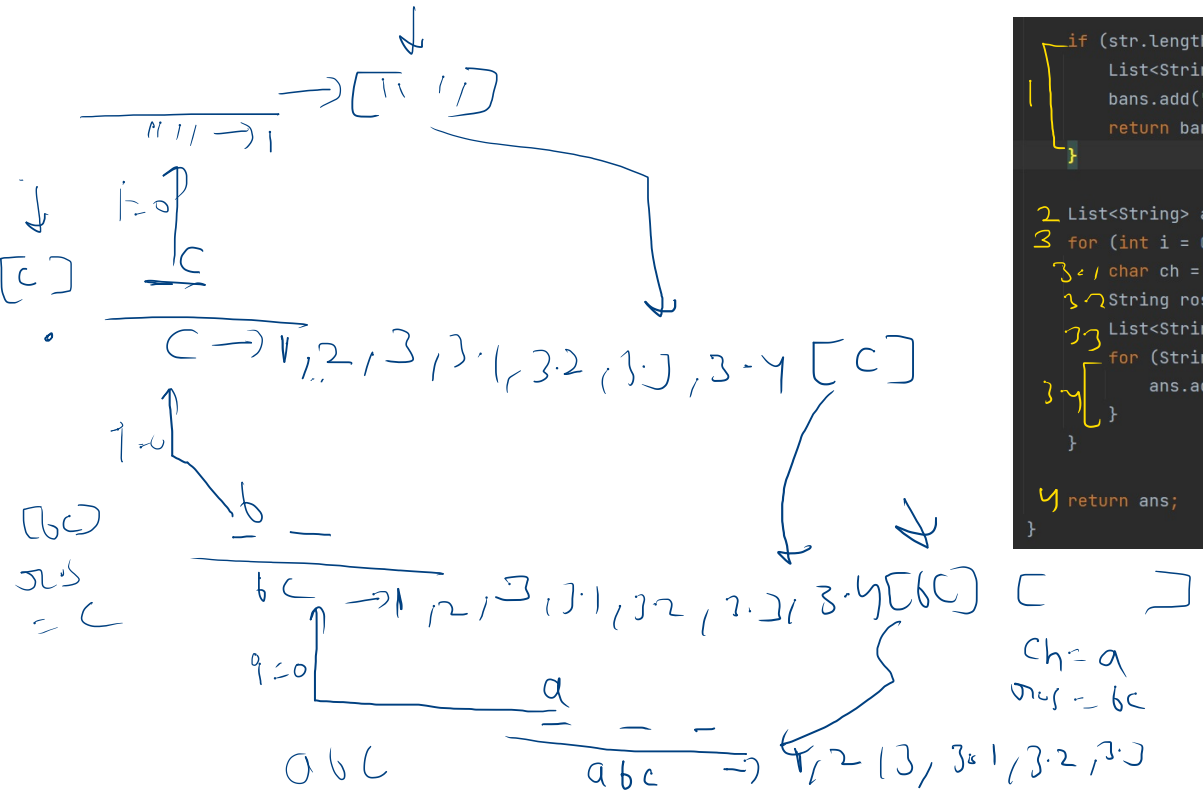
Sample Output 0

abc
acb
bac
bca
cab
cba




b
c
a
c
a
b
b
c
a
c
a
b
bc
ac
ab
a
b
c
abc


1
b c
c b
2
a c
c a
a c
3
a b
b a
a b
c
b
a
b c
a
a b c
a b c
a c b
b c a
c a b
c b a



```java
if (str.length() == 0) {
    List<String> bans = new ArrayList<>();
    bans.add("");
    return bans;
}

List<String> ans = new ArrayList<>();
for (int i = 0; i < str.length(); i++) {
    char ch = str.charAt(i);
    String ros = str.substring(0, i) + str.substring(i + 1);
    List<String> recans = solution(ros);
    for (String val: recans) {
        ans.add(ch + val);
    }
}

return ans;
}
```

Print Encodings (Day 26)

1 → a
2 → b
3 → c

25 - y
26 - z

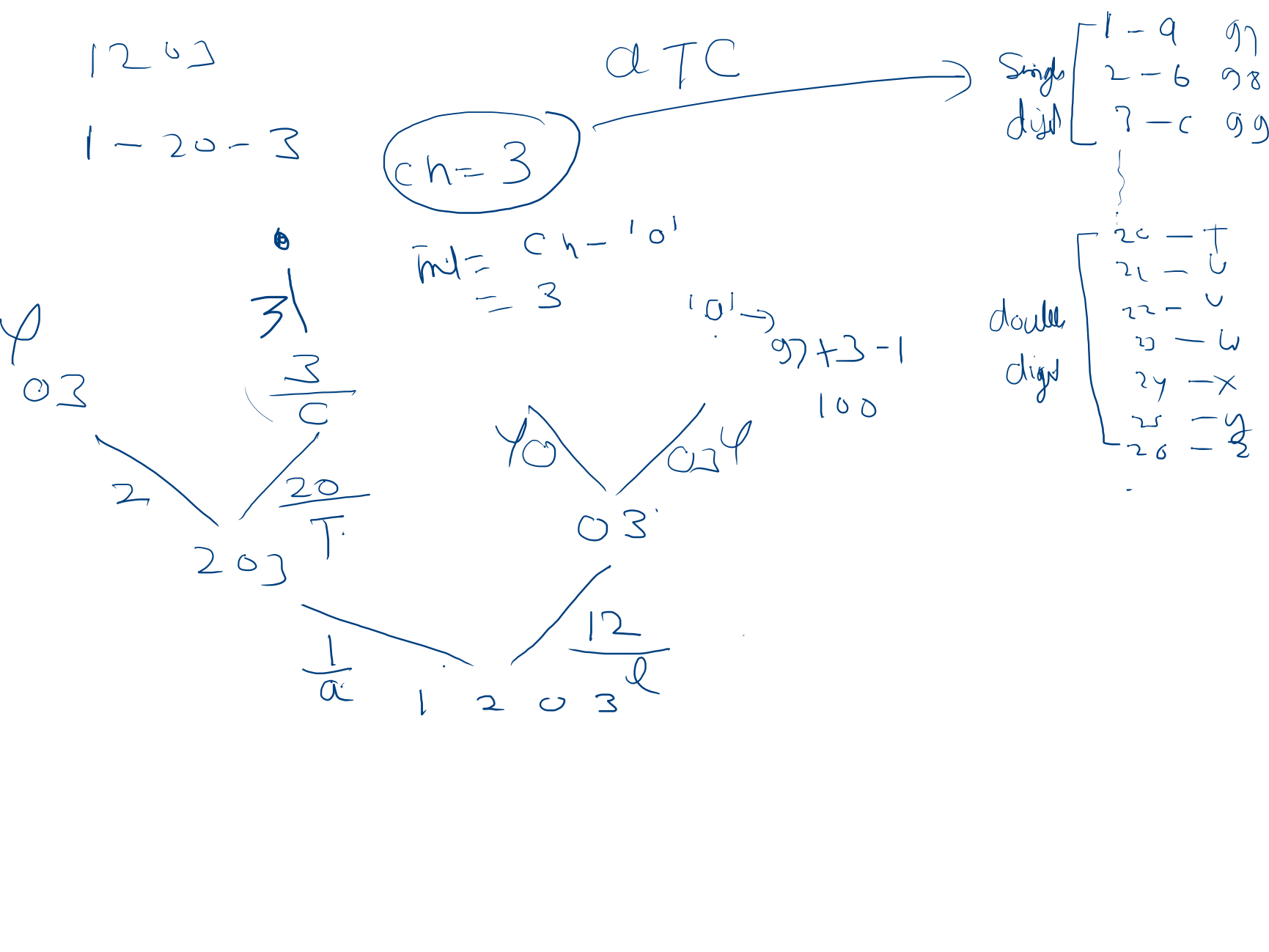
Ex: 1 2 3

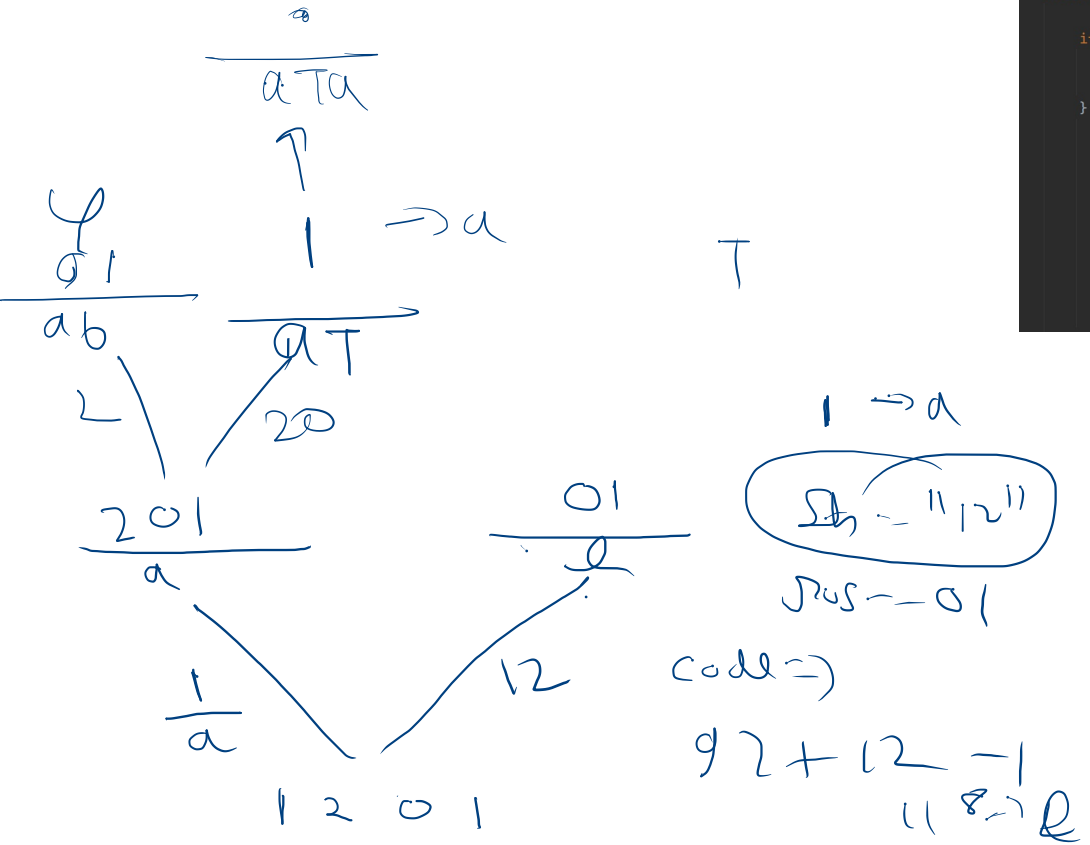
1 - 2 - 3
a b c

1 - 23
a - w

12 - 3
L C

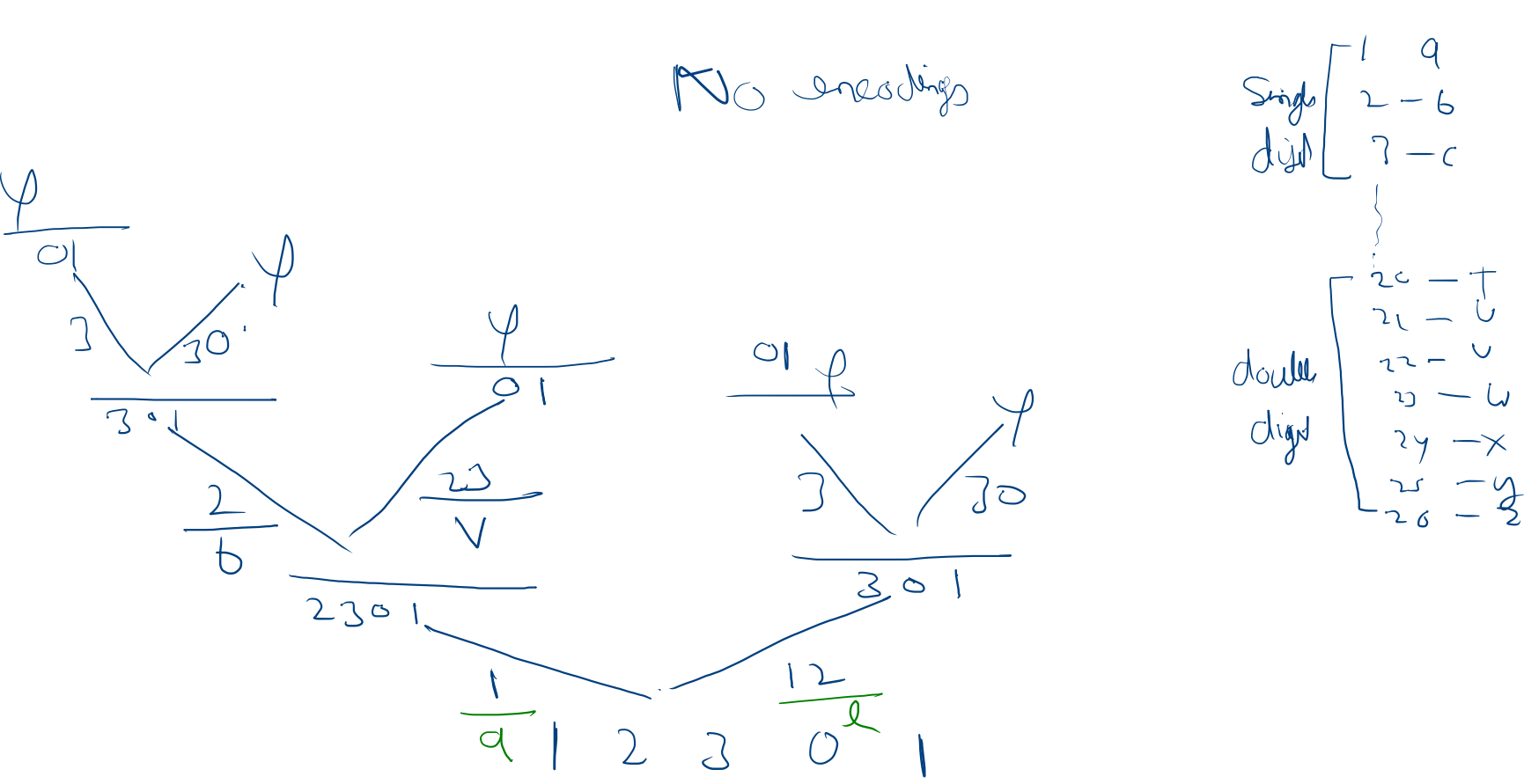
Ex: 013 → invalid as

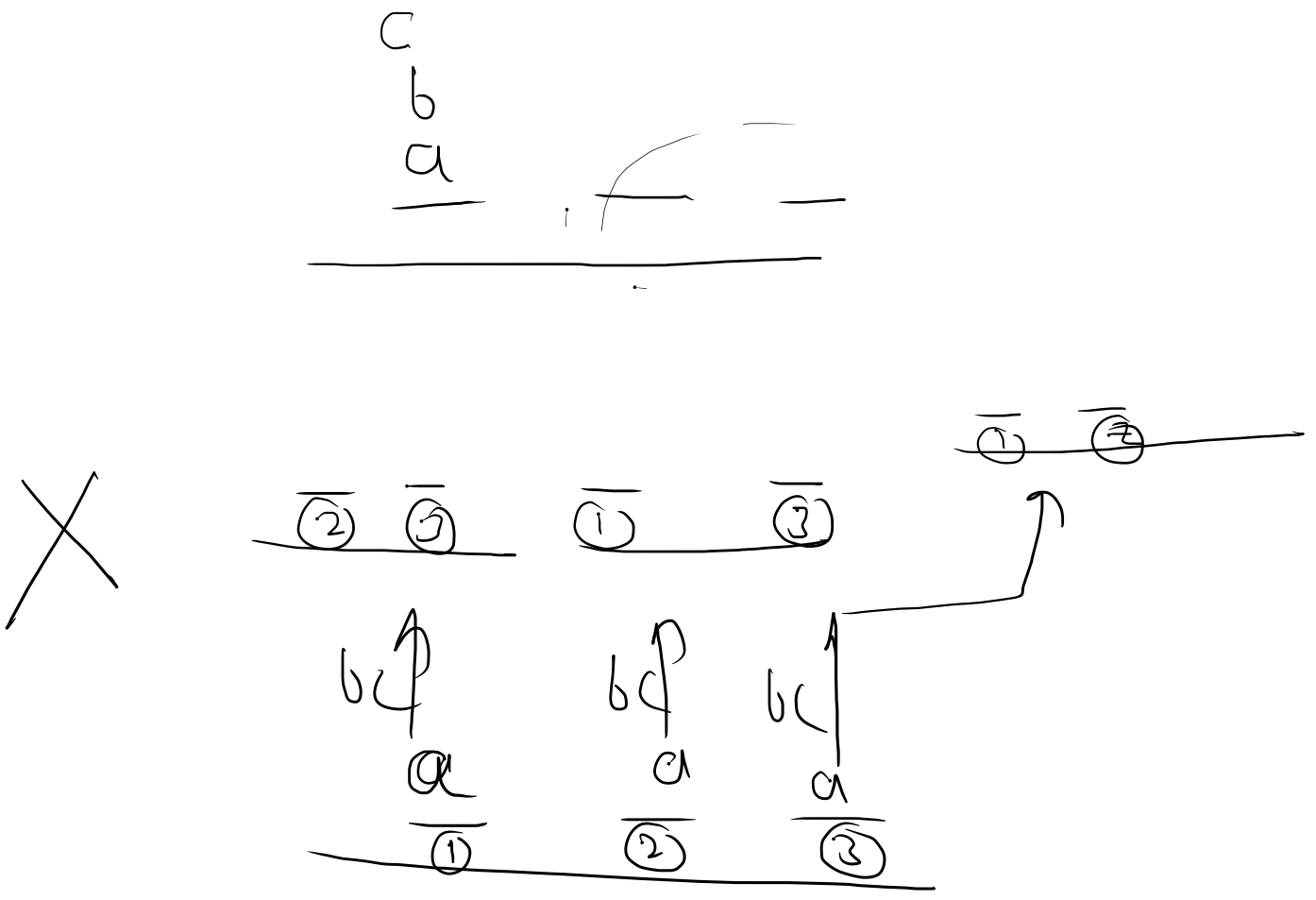

1203
1-20-3
ch=3
ATC
Single digit
1-9 97
2-6 98
7-c 99
double digit
20-t
21-u
22-v
23-w
24-x
25-y
26-z
ind= ch-'0'
=3
'0' → 97+3-1
100
03
2
20
T
1
a
12
l
1203



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

    if (str.length() == 0) {
        System.out.println(ans);
        return;
    } else if (str.length() == 1) {
        char ch = str.charAt(0);
        if (ch == '0') {
            return;
        } else {
            String ros = str.substring(1);
            int chInt = ch - '0';
            String code = (char)((int)('a') + chInt - 1) + "";
            solution(ros, ans: ans + code);
        }
```

```java
    } else { // string len ≥ 2

        // first char pick
        char ch = str.charAt(0);
        if (ch == '0') {
            return;
        } else {
            String ros = str.substring(1);
            int chInt = ch - '0';
            String code = (char)((int)('a') + chInt - 1) + "";
            solution(ros, ans: ans + code);
        }

        // two char pick
        String twoChar = str.substring(0, 2);
        String twoStrRos = str.substring(2);

        String code = (char) ((int)('a') + Integer.parseInt(twoChar) - 1) + "";

        if (Integer.parseInt(twoChar) ≤ 26) {
            solution(twoStrRos, ans: ans + code);
        }

    }
```


No encodings
Single digit
1 9
2 - 6
7 - c
Double digit
2c - t
21 - u
22 - v
23 - w
24 - x
25 - y
26 - z
1 2 3 0 1


c
b
a
1 2
2 3 1 3
b c b c b c
a a a
1 2 3