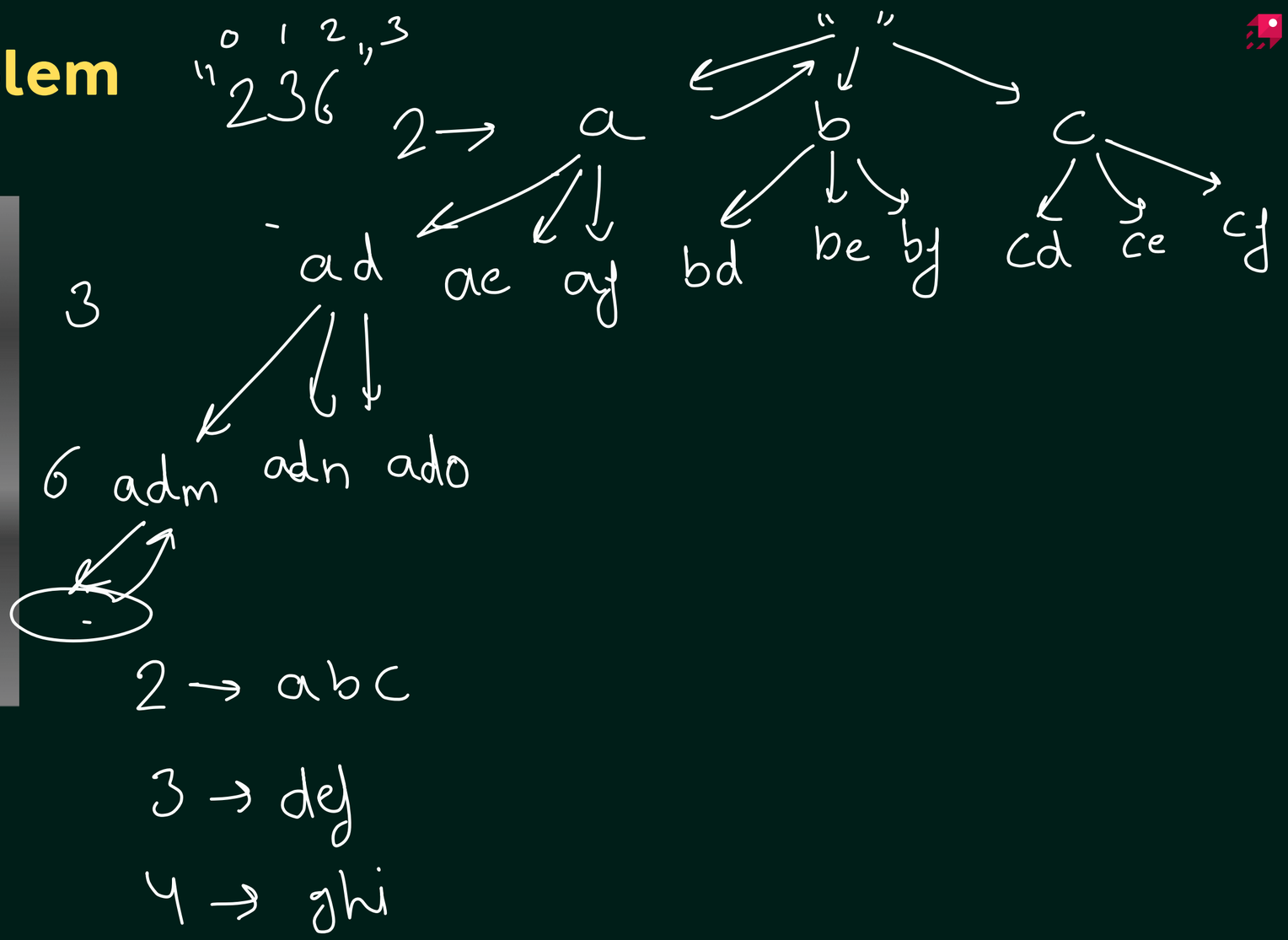
Week 4 LIVE

Backtracking Problems LIVE - 2

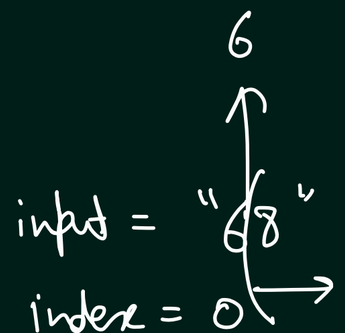
In This Lecture

1. Smart Keypad Problem
2. Palindromic Partitioning Problem

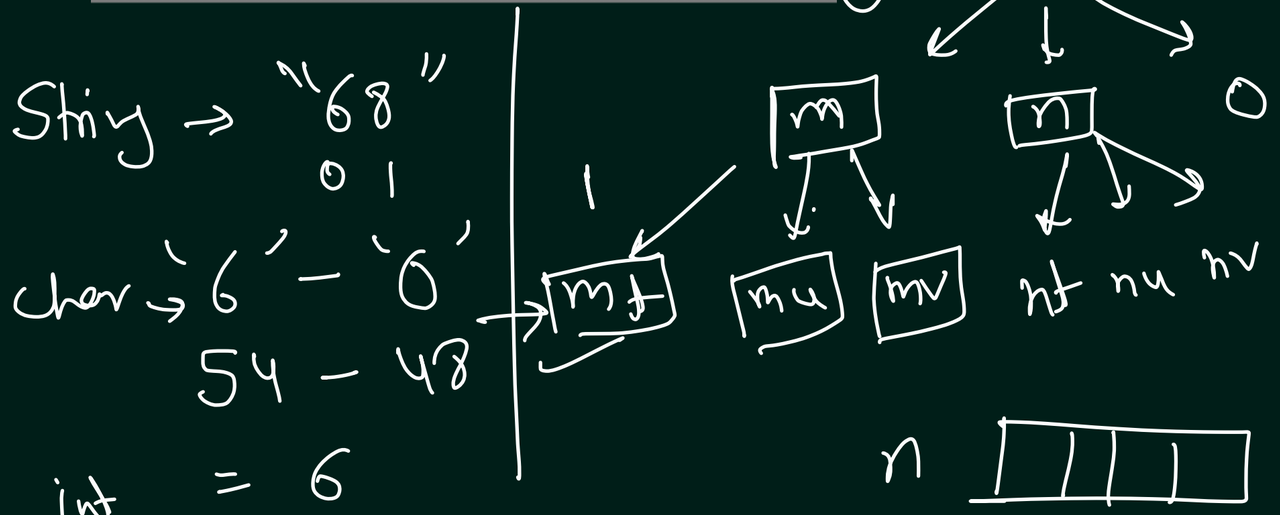
Smart Keypad Problem



Smart Keypad Problem

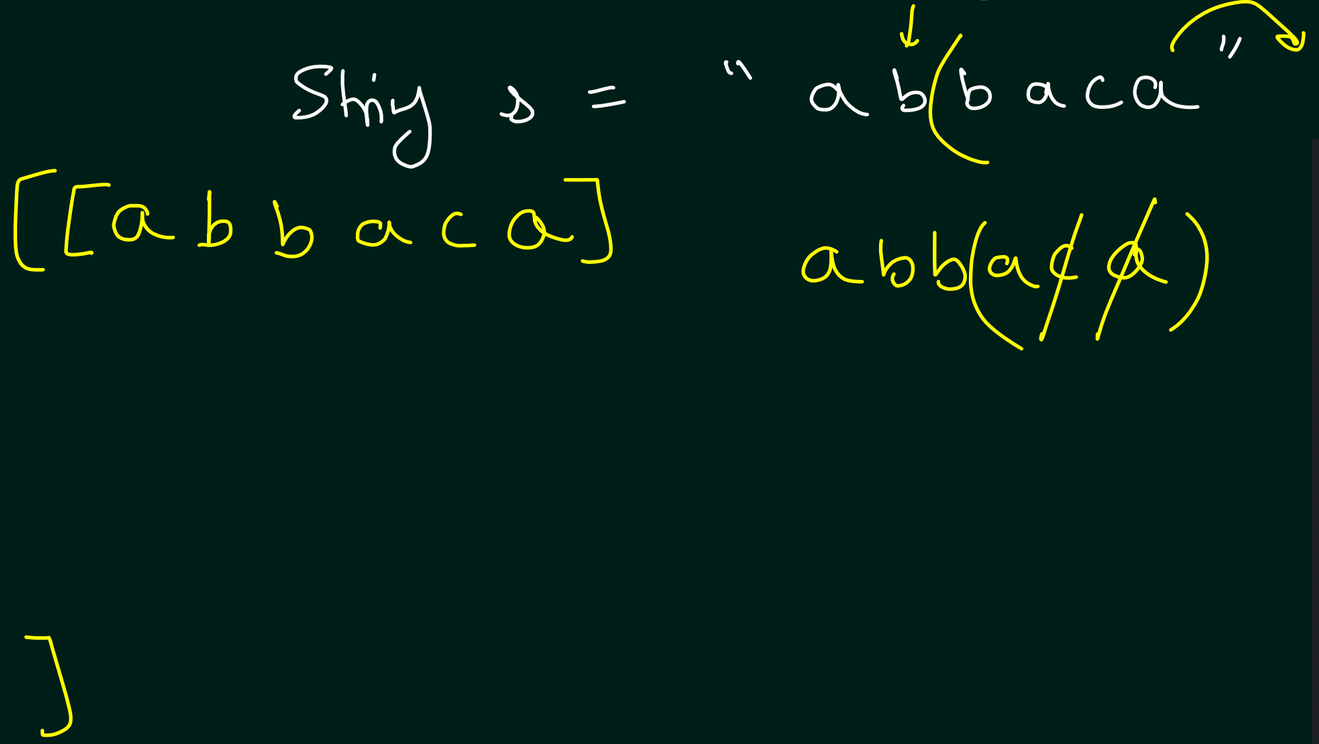



```
static void smartKeypadHelper(String input, int index, List<String> ans,
                              String a[], String cur) {
    if(index == input.length()) {
        ans.add(cur);
        return;
    }

    int indexOfButton = input.charAt(index)-'0';
    String buttonString = a[indexOfButton];

    for(int i = 0; i<buttonString.length(); i++) {
        cur = cur + buttonString.charAt(i);
        smartKeypadHelper(input, index+1, ans, a, cur);
        cur = cur.substring(0, cur.length()-1); // backtracking

        smartKeypadHelper(input, index: index+1, ans, a, cur: cur+buttonString.charAt(i));

    }

}
```



$O(k^n)$

Palindromic Partitioning Problem




```
static void palindromicHelper(String s, List<List<String>> ans,
                              int index, List<String> cur) {
    if(index == s.length()) {
        List<String> copyCur = new ArrayList<>(cur);
        ans.add(copyCur);
        return;
    }

    for(int i = index; i<s.length(); i++) {
        if(isPalindrome(s, index, i)) {
            cur.add(s.substring(index, i+1));
            palindromicHelper(s, ans, index: i+1, cur);
            cur.remove( index: cur.size()-1);
        }
    }
}
```

Palindromic Partitioning Problem

Sudoku

[0, 1, 3, 0, 6,],
[ ],