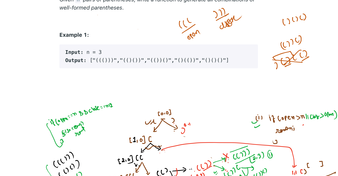
Lec-17
20240628 - 11:07PM

```
public static int count(String ques, String ans) {
    if (ques.length() == 0) {
        System.out.println(ans + " ");
        return 1;
    }
    char ch = ques.charAt(0);
    int ro0ount(ques.substring(1), ans);
    int ro1ount(ques.substring(1), ans + ch);
    return (ro0 + ro1);
}
```

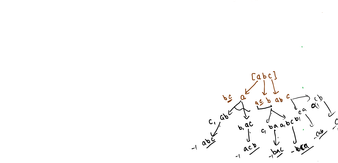
Given $n$ pairs of parentheses, write a function to generate all combinations of well-formed parentheses.

**Example 1:**

**Input:** n = 3
**Output:** ["((()))","(()())","(())()","()(())","()()()"]



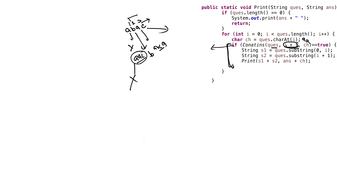
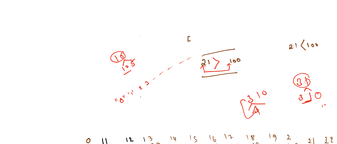
```
public static void Parentheses(int open, int close, String ans) {
    if (open == 0 && close == 0) {
        System.out.println(ans + " ");
        return;
    }
    if (open > 0) {
        Parentheses(n, open - 1, close, ans + "(");
    }
    if (close > open && close > 0) {
        Parentheses(n, open, close - 1, ans + ")");
    }
}
```



```
public static void Print(String ques, String ans) {
    if (ques.length() == 0) {
        System.out.println(ans + " ");
        return;
    }
    for (int i = 0; i < ques.length(); i++) {
        char ch = ques.charAt(i);
        String s1 = ques.substring(0, i);
        String s2 = ques.substring(i + 1);
        Print(s1 + s2, ans + ch);
    }
}
```

```
public static void Print(String ques, String ans) {
    if (ques.length() == 0) {
        System.out.println(ans + " ");
        return;
    }
    for (int i = 0; i < ques.length(); i++) {
        char ch = ques.charAt(i);
        if (Counting(ques, ch) == 1 && ans.contains(ch) == true) {
            String s1 = ques.substring(0, i);
            String s2 = ques.substring(i + 1);
            Print(s1 + s2, ans + ch);
        }
    }
}
```

```
public static void Counting(int curr, int n) {
    if(curr==0 ||
        return;
    }
    System.out.println(curr);
    int i = 0;
    if (curr == 0) {
        i = 1;
    }
    for (; i <= 9; i++) {
    }
}
```



```
public static void Counting(int curr, int n) {
    if (curr > n) {
        return;
    }
    System.out.println(curr);
    int i = 0;
    if (curr == 0) {
        i = 1;
    }
    for (; i <= 9; i++) {
        Counting(curr*10+i, n);
    }
}
```