CODING SHUTTLE

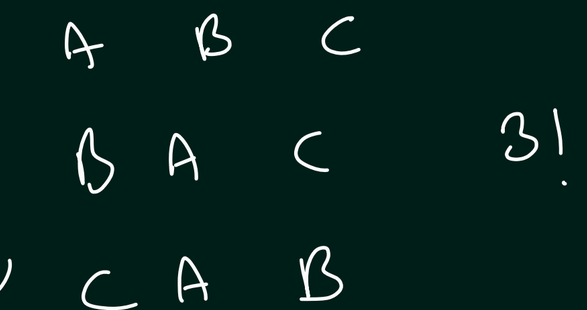
Backtracking - 2

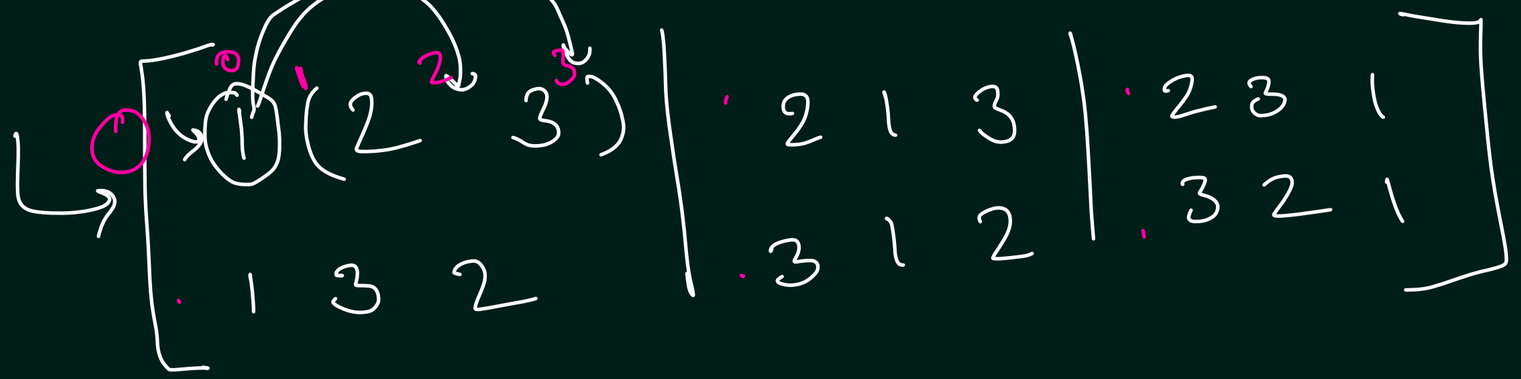
In This Lecture

1. Permutations of an Array
2. Generate Parentheses

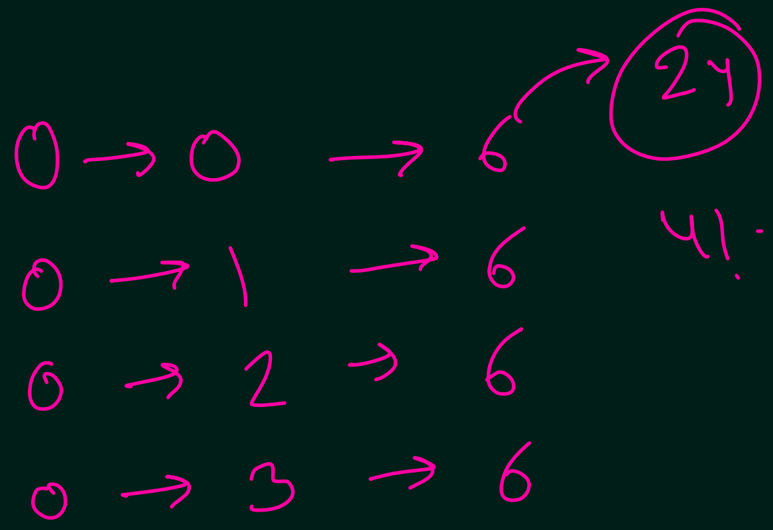
Permutations of an Array

Input: nums = [1,2,3]
Output: [[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]

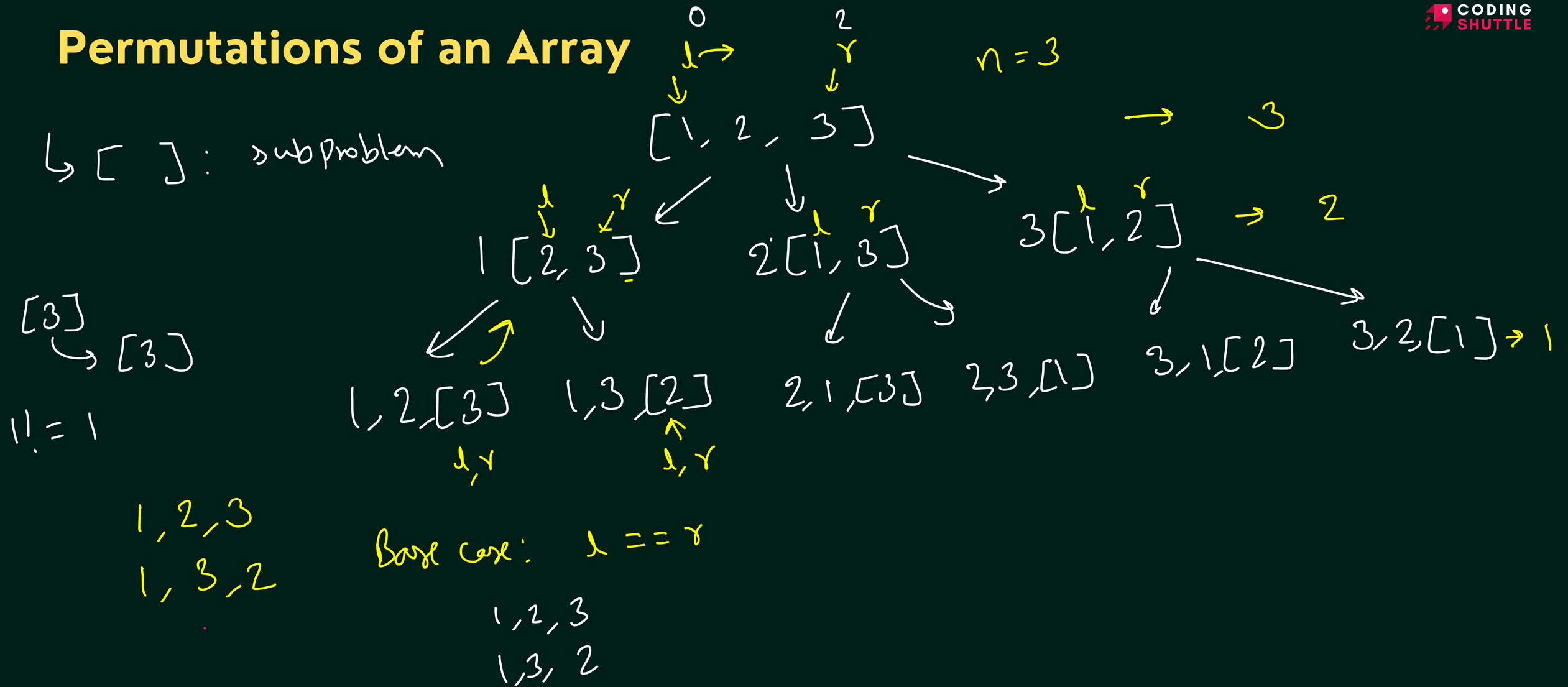




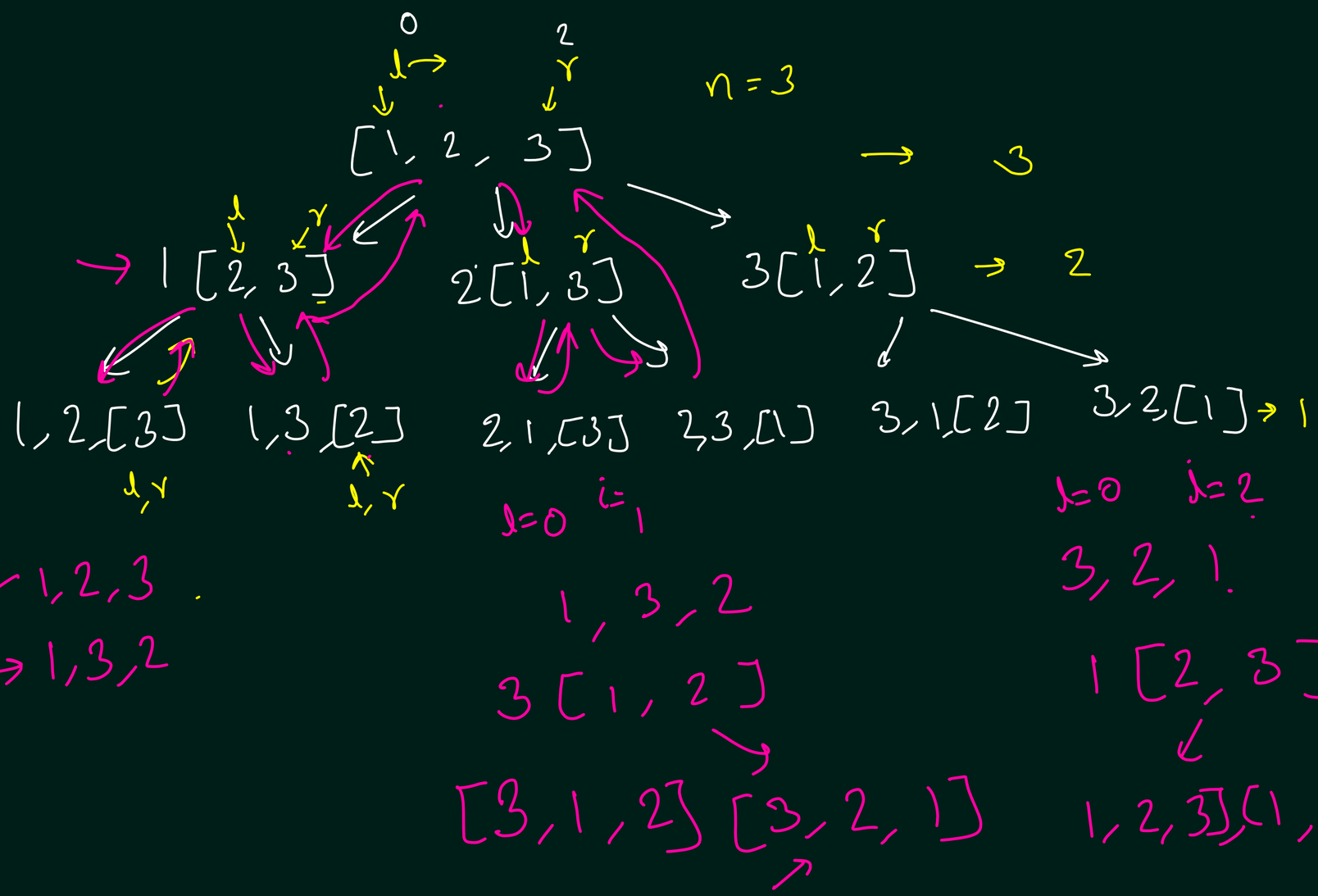
Input: nums = [0,1]
Output: [[0,1],[1,0]]



Permutations of an Array



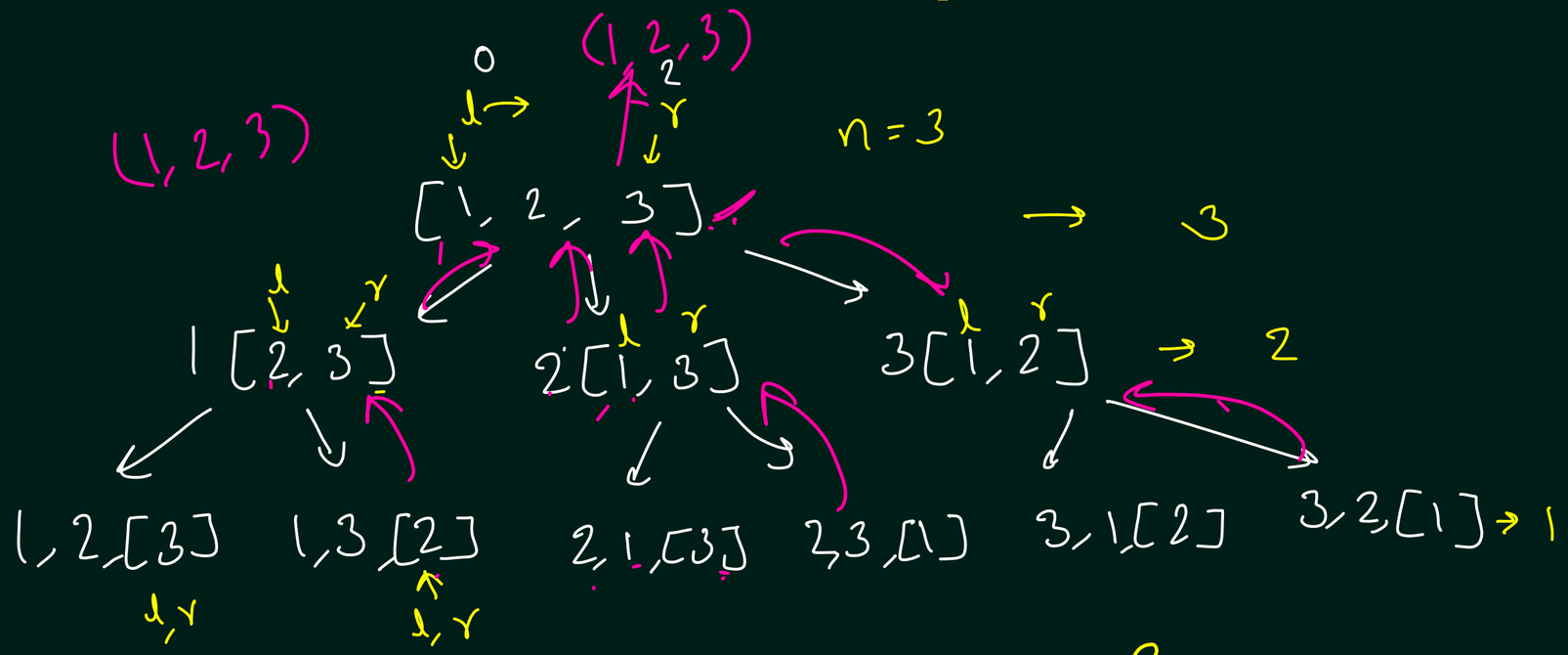
Permutations of an Array




```java
public static void main(String[] args) {
    int a[] = {1, 2, 3};
    permute(a, l: 0, r: a.length-1);
}

2 usages
static void permute(int a[], int l, int r) {
    if(l == r) {
        printArray(a);
        return;
    }
    for(int i = l; i<=r; i++) {
        swap(a, i, l);
        permute(a, l: l+1, r);
        //swap(a, i, l); //Backtracking
    }
}
```

Permutations of an Array




```java
public static void main(String[] args) {
    int a[] = {1, 2, 3};
    permute(a, l: 0, r: a.length-1);
}

2 usages
static void permute(int a[], int l, int r) {
    if(l == r) {
        printArray(a);
        return;
    }
    for(int i = l; i<=r; i++) {
        swap(a, i, l);
        permute(a, l: l+1, r);
        //swap(a, i, l); //Backtracking
    }
}
```

1, 2, 3
1, 3, 2
2, 1, 3
2, 3, 1
3, 1, 2
3, 2, 1

$O(n! * n)$

n = 3 → 3!

$O(n! * n)$

$O(n)$

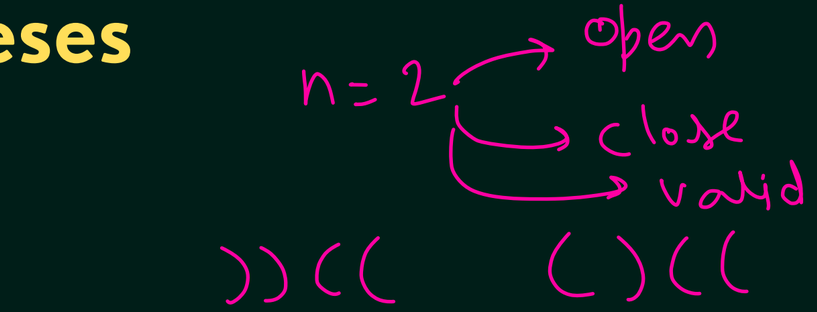
Generate Parentheses

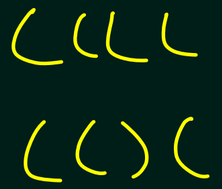
For n = 2

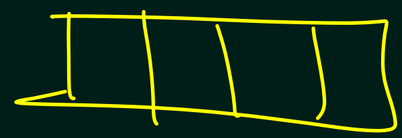
Print: (()), ()()

For n = 3

Print: ((())), (()()), (())(), ()(()), ()()()



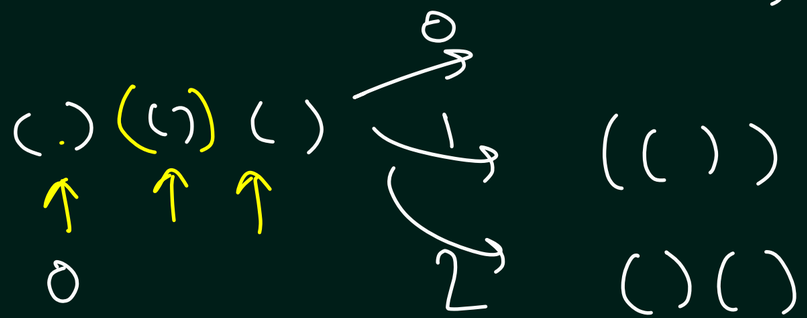




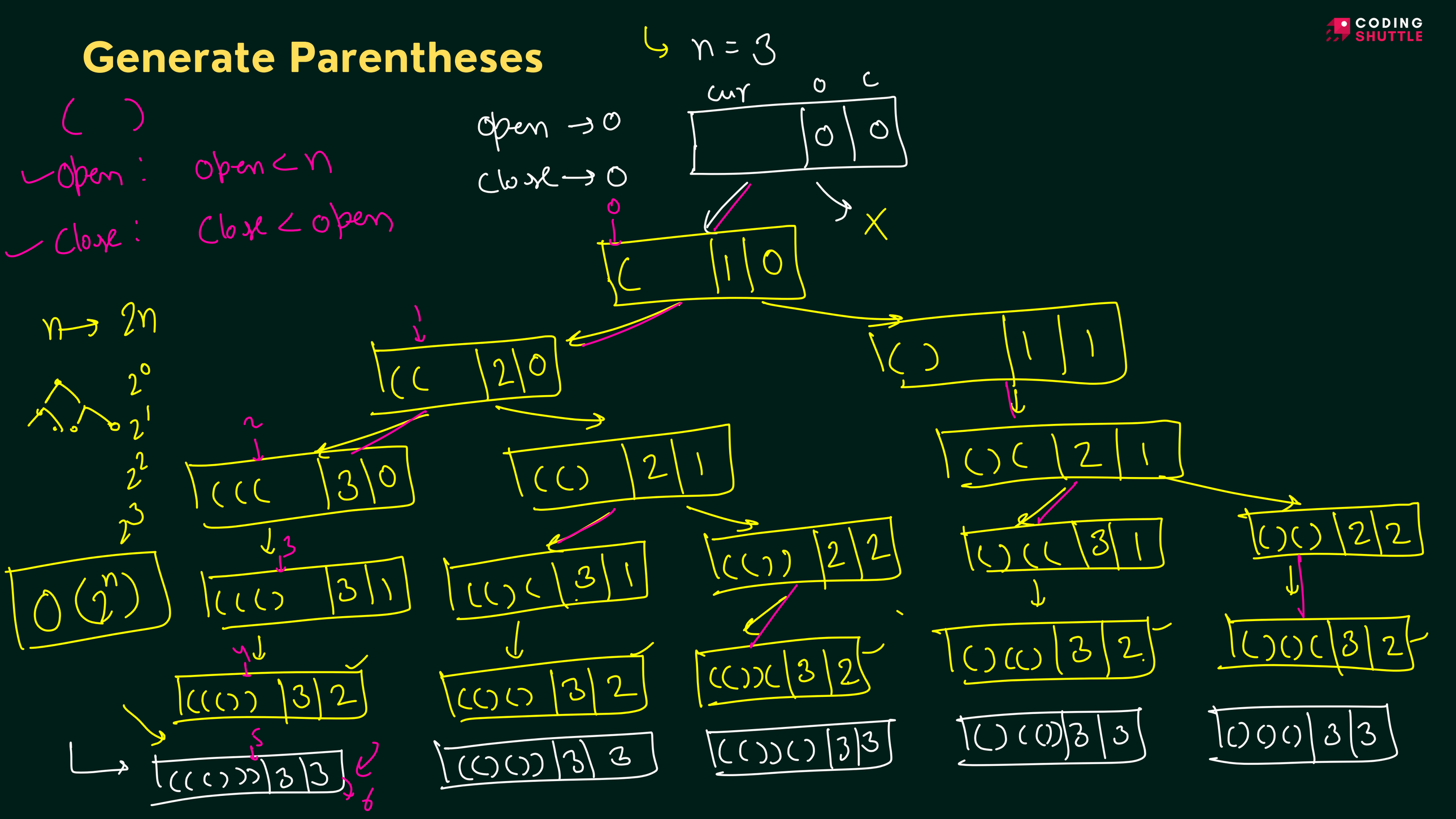
$2^4 \rightarrow 16$

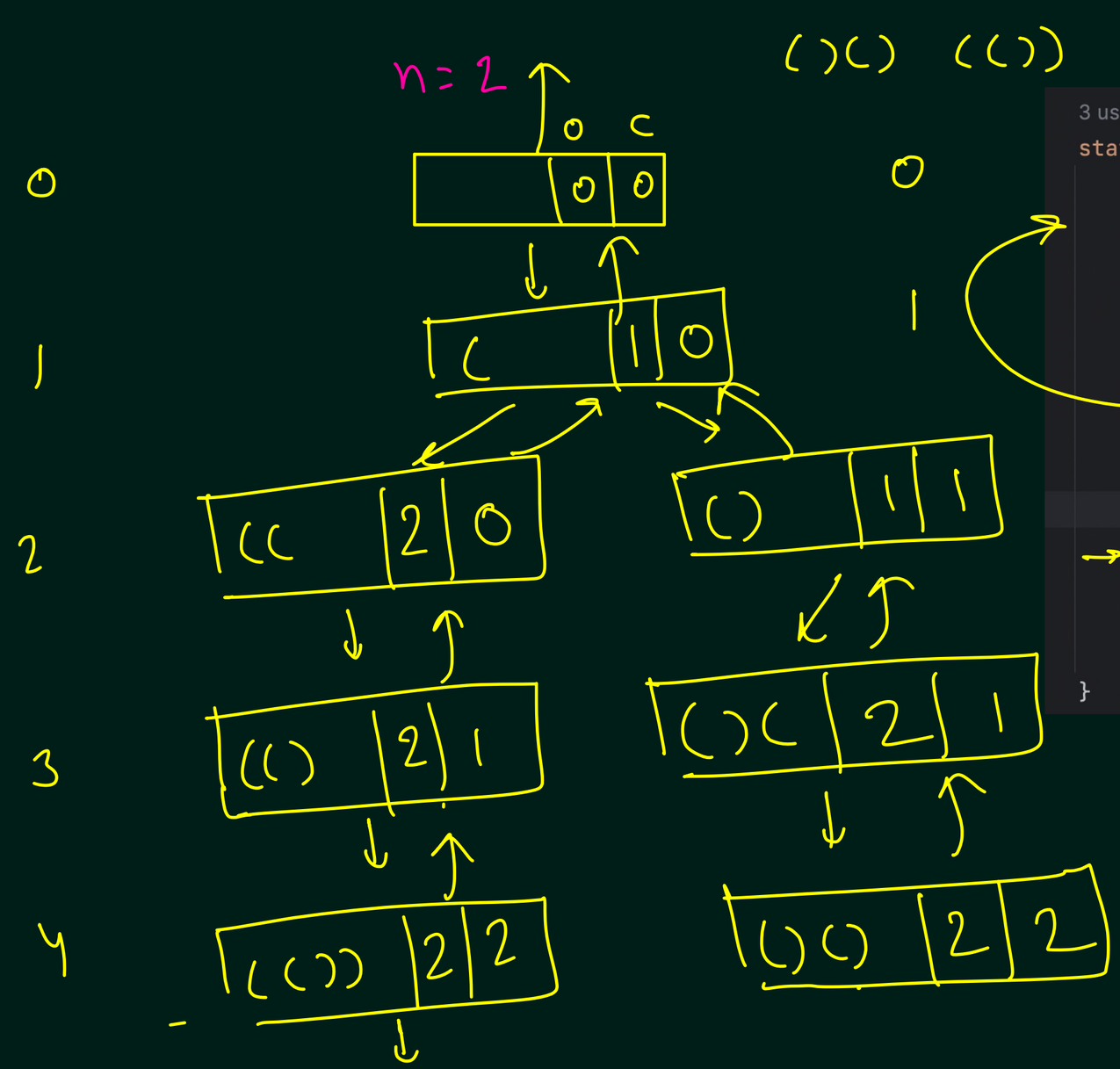
$O(n * 2^n)$

$O(2^n)$



Generate Parentheses






```
3 usages                                                    2    ""    0    0    0
static void generateParentheses(int n, String cur, int i, int open, int close) {
    if(i == 2*n) {
        System.out.println(cur);
        return;
    }
    if(open < n) {
        cur = cur + '(';
        generateParentheses(n, cur, i: i+1, open: open+1, close);
        cur = cur.substring(0, cur.length()-1); // backtracking
    }

    if(close < open) {
        cur = cur + ')';
        generateParentheses(n, cur, i: i+1, open, close: close+1);
    }
}
```