### **L1 (Recursion Fundamentals)**

1. Calculating Factorials
2. Calculating Fibonacci Numbers
3. Fast Exponentiation

### **L2 (Problem Solving on Recursion)**

1. <https://practice.geeksforgeeks.org/problems/help-the-old-man3848/1#_=_>
2. <https://www.codechef.com/problems/KCHAR/>
3. <https://www.codechef.com/INOIPRAC/problems/INOI1502/>
4. <https://www.hackerearth.com/practice/basic-programming/recursion/recursion-and-backtracking/practice-problems/algorithm/gcd-strings/>

### **L3 (Backtracking)**

1. <https://leetcode.com/problems/permutations/>
2. <https://leetcode.com/problems/subsets/>
3. <https://www.hackerearth.com/practice/basic-programming/recursion/recursion-and-backtracking/practice-problems/algorithm/n-queensrecursion-tutorial/>
4. <https://leetcode.com/problems/n-queens/>

### **L5 (Problem Solving on Backtracking)**

1. Travelling Salesman Problem
2. <https://practice.geeksforgeeks.org/problems/subset-sum-problem2014/1>
3. Print all possible paths from top left to bottom right of a n\*n matrix covering every cell exactly once.
4. <https://www.geeksforgeeks.org/word-break-problem-using-backtracking/>

### **L6 (Dynamic Programming: Introduction)**

1. <https://leetcode.com/problems/fibonacci-number/>
2. <https://leetcode.com/problems/coin-change/>
3. <https://www.codechef.com/ZCOPRAC/problems/ZCO14002>
4. <https://codeforces.com/problemset/problem/72/G>
5. <https://codeforces.com/problemset/problem/797/B>
6. <https://codeforces.com/problemset/problem/1282/B1>
7. <https://codeforces.com/problemset/problem/1472/C>
8. <https://codeforces.com/problemset/problem/166/E>

### 

### **L7 (Dynamic Programming: Classical Problems - 1)**

1. <https://leetcode.com/problems/unique-paths/>
2. <https://leetcode.com/problems/unique-paths-ii/>
3. <https://leetcode.com/problems/longest-increasing-subsequence/>
4. <https://practice.geeksforgeeks.org/problems/longest-common-substring1452/1>
5. <https://leetcode.com/problems/longest-common-subsequence/>