Algorithm Design

Resources:

1. hiredintech
2. interview cake
3. mit
4. Algo
5. coding Qs
6. time complexity
7. topcoder time complexity 1
8. topcoder time complexity 2
9. dynamic programming
10. daily coding problem
11. geekforgeeks, leetcode, topcoder, codeforces, interview bit, bytesforbytes

Algorithm design canvas:

1. Constraints
    1. input
    2. edge cases
    3. contraints handout
2. Ideas
    1. start thinking about a simpler version of the problem and draw some conclusions about how to solve the original problem.
    2. try few examples

    3. suitable data structures
3. Complexities
    1. time
    2. memory
    3. <u>common complexities handout</u>
    4. O(2^N) → growth doubles with each addition to the input data set
    5. O(logN) → input data set divided by 2 in each iteration → iterative halving of data sets
    6. O(n!) → permutation
4. Code
5. Tests
    1. edge cases → 0, negative numbers, duplicates, empty arrays, empty strings, etc.
    2. Cases where there's no solution
    3. non-trivial functional test cases