Algorithms Paradigms

Different problem can be solved using different style of approaches.
For example, If we are re-calculating a result. It is always beneficial to store result in order to prevent recalculation and thus save time.
Or we might go with the solution which seem right for now
Or we might divide our problem into sub-problems and solve those sub-problems instead
Or We just calculate all the possible results and see the answer

As per our example, we will be discussing 4 major algorithm paradigms i.e.

1. **Brute Force or Complete Search Technique**
   In this method we generate possible solutions and then we find out solution among them.
   Sometimes, brute force is enough for Normal div2A and div2B questions
2. **Divide and Conquer**
   We divide our problem into parts, pass them into functions and then ask those functions to solve them And then derive a solution from solution of those subproblems
3. **Greedy Alogrithms**
   Let's Take a simple example of finding a global maxima or minima in a graph. If we think of our idea as finding a point with $\frac{dy}{dx}f(x) = 0$, then we might get a wrong answer since, the point found maybe local maxima but not global maxima
4. **Dynamic Programming**
   This is the most interesting paradigm to solve problems. Here save our previous results in order to prevent recalculation of some results again and again.
   This paradigm is so powerful that we can bring time complexity of $O(2^n)$ down to polynomial complexity