Algorithmic Techniques

There are three main algorithmic techniques: Divide and conquer, greedy algorithms, and dynamic programming.

1. **Divide and Conquer.** In this approach, you find a way to divide the problem into pieces such that: if you recursively solve each piece, you can stitch together the solutions to each piece to form the overall solution. Both Merge Sort and QuickSort are classic examples of divide-and-conquer algorithms. Another famous example is modular exponentiation (used in cryptography).

2. **Greedy Algorithms.** In a greedy algorithm, the optimal solution is built up one piece at a time. At each stage the best feasible candidate is chosen as the next piece of the solution. There is no back-tracking. An example of a greedy algorithm is Huffman coding. Another famous example is several algorithms for finding a minimum spanning tree of a graph.

3. **Dynamic Programming.** If you find a way to break the problem into pieces, but the number of pieces seems to explode, then you probably need the technique known as dynamic programming. We do not study this.