Dynamic Programming

Dynamic Programming

- **Dynamic programming like the divide and conquer method, solves problem by combining the solutions of sub problems**

- **Divide and conquer method partition the problem into independent sub problems, solves the sub problems recursively and then combine their solutions to solve the original problem.**

Dynamic Programming

- **Dynamic programming is applicable, when the sub-problems are NOT independent, that is when sub-problems share sub sub-problems.**
- **It is making a set of choices to arrive at optimal solution.**
- **If sub-problems are not independent, we have to further divide the problem.**
- **In worst case, we may end-up with an exponential time algorithm.**

Dynamic Programming

- **Frequently, there is a polynomial number of sub-problems, but they get repeated.**
- **A dynamic programming algorithm solves every sub-problem just once and then saves its answer in a table, thereby avoiding the work of re-computing the answer every time the sub-problem is encountered**
- **So we end up having a polynomial time algorithm.**
- **Which is better, Dynamic Programming or Divide & conquer?**

Optimization Problems

- **Dynamic problem is typically applied to <u>Optimization Problems</u>**

- **In optimization problems there can be many possible solutions. Each solution has a value and the task is to find the solution with the optimal (Maximum or Minimum) value. There can be several such solutions.**

4 steps of Dynamic Programming Algorithm

1. Characterize the structure of an optimal solution.

2. Recursively define the value of an optimal solution.

3. Compute the value of an optimal solution bottom-up.

4. Construct an optimal solution from computed information

Often only the value of the optimal solution is required so step-4 is not necessary.