**Dynamic Programming**

Dynamic programming was invented by U.S. mathematician Richard Bellman in 1950. Like greedy algorithms, it is also used to solve optimization problems. But unlike greedy approach, dynamic programming always ensures optimal / best solution.

A feasible solution is a solution that satisfies constraints of the problem. When the problem has multiple feasible solutions with different cost, the solution with the minimum cost or maximum profit is called optimal solution

Cost metric depends on the problem. For sorting problem, cost metric may be a number of comparisons or number of swaps. For matrix multiplication, a cost metric is a number of multiplications. For knapsack problem, cost metric is total profit earned.

**General Strategy**

Dynamic programming is powerful design technique for optimization problems. Here word “programming” refers to planning or construction of a solution, it does not have any resemblance with computer programming.

Divide and conquer divides the problem into small sub problems. Sub problems are solved recursively. Unlike divide and conquer, sub problems in dynamic programming are not independent. Sub problems in it overlap with each other. Solutions of sub problems are merged to get the solution of the original large problem.

In divide and conquer, sub problems are independent and hence repeated problems are solved multiple times. Dynamic programming saves the solution in the table, so when the same problem encounters again, the solution is retrieved from the table. It is bottom up approach. It starts solving the smallest possible problem and uses a solution of the smaller problem to build solution of the larger problem.

**Limitations**

* The method is applicable to only those problems which possess the property of principle of optimality.
* We must keep track of partial solutions.
* Dynamic programming is more complex and time-consuming

**Control Abstraction**

Dynamic programming (DP) splits the large problem at every possible point. When the problem becomes sufficiently small, DP solves it.

Dynamic programming is bottom up approach, it finds the solution of the smallest problem and constructs the solution of the larger problem from already solved smaller problems.

To avoid recomputation of the same problem, DP saves the result of sub problems into thetable. When next time same problem encounters, the answer is retrieved from the table by lookup procedure.

**Control abstraction for dynamic programming is shown below:**

Algorithm DYNAMIC\_PROGRAMMING (P)

if solved(P) then

return lookup(P)

else

Ans ← SOLVE(P)

store (P, Ans)

end

FUNCTION SOLVE(P)

if sufficiently small(P) then

solution(P) // Find solution for sufficiently small problem

else

Divide P into smaller sub problems P1, P2, ..., Pn

Ans1 ← DYNAMIC\_PROGRAMMIN(P1)

Ans2 ← DYNAMIC\_PROGRAMMIN(P2)

.

.

Ansn ← DYNAMIC\_PROGRAMMIN(Pn)

return (combine(Ans1, Ans2 ..., Ansn))

end

**Characteristics of Dynamic Programming**

Dynamic programming works on following principles:

* Characterize structure of optimal solution, i.e. build a mathematical model of the solution.
* Recursively define the value of the optimal solution.
* Using bottom up approach, compute the value of the optimal solution for each possible sub problems.
* Construct optimal solution for the original problem using information computed in the previous step.

**Applications of Dynamic Programming**

Dynamic programming is used to solve optimization problems. It is used to solve many real life problems such as,

* Make a change problem
* Knapsack problem
* Optimal binary search tree
* Travelling salesman problem
* All pair shortest path problem
* Assembly line scheduling
* Multi stage graph problem

**Principle of Optimality**

Principle of optimality: “In an optimal sequence of decisions or choices, each sub sequence must also be optimal”.

The principle of optimality is the heart of dynamic programming. It states that to find the optimal solution of the original problem, a solution of each sub problem also must be optimal. It is not possible to derive optimal solution using dynamic programming if the problem does not possess the principle of optimality.