**Module 3 Notes – Dynamic Programming**

* **Dynamic programming** solves problems by combining the solutions to subproblems. Divide-and-conquer algorithms partition the problem into disjoint subproblems, solve the problems recursively, and then combine their solutions to solve the original problem.
  + **Dynamic programming** applies when the subproblems overlap, when subproblems share subproblems
  + Some problems solved by **DP**:
    - Appear to be exponential but have a polynomial solution with DP
    - In many cases are optimization problems (min/max)
    - Defined by or formulated as recurrences with overlapping subproblems
    - Optimal solution to a problem contains optimal solutions to subproblems.
* We typically apply **dynamic programming** to ***optimization problems***, problems that can have many possible solutions. Each solution has a value and we wish to find a solution with the optimal (max or min) value.
* When developing **dynamic-programming algorithm**, we follow these four steps:
  + Characterize the structure of an optimal solution
  + Recursively define the value of an optimal solution
  + Compute the value of an optimal solution, typically in a bottom-up fashion
  + Construct an optimal solution from computed information
    - If we need only the value of the optimal solution, and not the solution itself, then we can skip step 4
  + Simpler steps include…
    - define subproblems
    - guess part of the solution
    - relate subproblems to solutions
    - recurse + memoize or build a DP bottom-up table
    - solve the original problem
* The 2 conditions an optimization problem must have in order for dynamic programming to apply are:
  + optimal substructure
  + overlapping subproblems
* A problem exhibits ***optimal substructure*** if an optimal solution to the problems contains within it optimal solutions to subproblems.
* The running time of a **dynamic programming algorithm** depends on the product of two factors:
  + The number of subproblems overall
  + How many choices we look at for each subproblem