L1: What is Dynamic Programming

# Define & Demonstrate need

* Consider Dynamic Programming as a black box that takes in a problem statement as an input such that it produces a solution as an output.
* It helps solve problems in polynomial time where naïve approach would take exponential time. O(NC) [Polynomial time] and O(CN) [Exponential time]
* It is an optimization technique, to go from exponential to linear complexity

# Properties of DP problems

* **Optimal Substructure:** A problem has optimal substructure property when the optimal solution of the problem can be constructed from the optimal solution of the substructures of the given problem.
  + **Where,** 
    - **Optimal** means best or most favorable
    - **Substructure** sub problem of the main problem
  + Consider a problem X, we know that a complex problem can be systematically solved by breaking it down. Consider the fragment solved first to be X­‑1, X2, X3 … Xn.
  + In the above case we can first solve X1, then use the results of X1 to solve for X2 sequentially without resolving X1 hence reducing the complexity.
* **Overlapping Sub problems:** When you break a problem into sub problems into sub problems you will notice that you need to re-calculate some work multiple times. Consider the Fibonacci example, in which the colored boxes in the following diagram highlight recalculation of a given sub problem multiple times.

L2: Elementary Problem

# How to identify if the problem can be solved with Dynamic Programming

* There are two sub classes of dynamic programming problems, mentioned below:
  + **Combinatoric:** They answer the question***“How Many …?”*.** The end goal in such problems is to count something**.** Some common types of these questions are:
    - ***How many ways to make a change?***
    - ***How many ways to traverse a graph?***
    - ***How many steps needed to get from point A to B?***
  + **Optimization:** In these problems we need to find a strategy that maximizes or minimizes some function. Some common types of these questions are:
    - ***What is the maximum profit gained by buying and selling a stock?***
    - ***What is the minimum cost to travel from New York to Mumbai?***
* Dynamic programming relies on the fact that the optimal solution to the optimization problems or combinatorial problem depends on the optimal solution for its overlapping sub problems.

# Problem 1: Find the sum of the first N numbers

* Break the problem into multiple subproblem, the simplest possible subproblem is for N=1 in this case the smallest possible sum is 1. Hence f(N=1) = 1.
* Let’s progress to N=2**,** now we must calculate f(N=2). This problem can be written as f(N=1) + 2. This is how we devise a general formula for the elements.
* Hence the general problem for our problem statement is: f(N) = f(N-1) + N
* Code available at: <link1>