Top-Down Problem Solving

Remington Greko, Tyler Gutowski, and Spencer Hirsch

February 16, 2023

Work with your team to write a report showing your knowledge of the Recursion. Submit the team's report on Canvas. Include a task matrix indicat- ing who did what.

Recursion

There is a cute definition of recursion in the Hacker's Dictionary: Recursion: See Recursion

There is a good description of recursion on Wikipedia, read it. Top-down problem solving requires solving a recurrence relation. There are a similarities between recurrence equations and ordinary differential equations should you desire to explore.

After successful completion of this quest you will understand how to model the time complexity of a recursive algorithm by a recurrence of the form

$$T(n) = aT(n/b) + f(n)$$

together with some initial conditions to get things started. Interpret the recurrence above as saying: To solve a problem with input size n, solve a problem of size n/b (you may need to do this a times) and apply a forcing function f(n) at each step.

There are many ways to solve a recurrence:

1. Guess or Given and Prove
2. Unrolling also called substitution
3. The Master Theorem
4. Generating Functions

Name	**Section**
Remington Greko	Second example of Dynamic Programming & Shortest Path
Tyler Gutowski	Third Example of Dynamic Programming & Eight Queens Problem
Spencer Hirsch	How Dynamic Programming Works, One Example use of Dynamic Programming & 0-1 Knapsack Problem algorithm example