Algorithms for Data Science

Optimization: Dynamic Programming

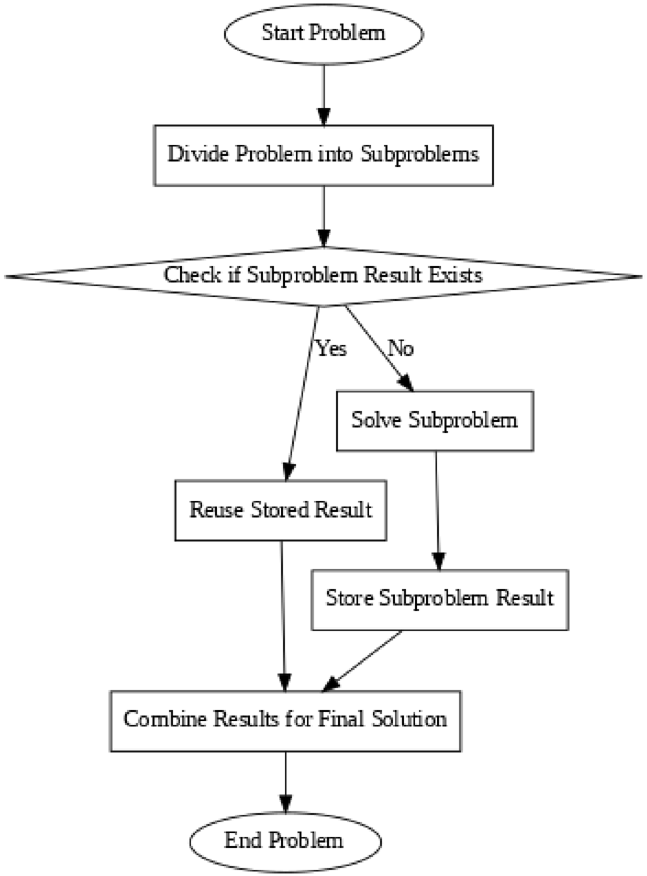
Johns Hopkins
Whiting School of Engineering

Dynamic Programming

Dynamic programming (DP) is a method for solving complex optimization problems by breaking them down into simpler, overlapping subproblems.

Key Idea

- Solve each subproblem once and store the results to avoid redundant calculations (memorization).
- Leverages optimal substructure, where the solution to a larger problem can be constructed from solutions to its subproblems.



Principles of DP

Optimal Substructure

- A problem exhibits optimal substructure if its solution can be constructed from the solutions of its subproblems.
- E.g. In the shortest path problem, the shortest path from A to C via B is the sum of the shortest paths from A to B and B to C.

Overlapping Subproblems

- A problem exhibits overlapping subproblems if the same smaller problem is solved multiple times.
- E.g. Computing Fibonacci numbers using recursion results in redundant calculations of the same Fibonacci values.

Memoization vs. Tabulation

- Memoization: Top-down approach using recursion and caching subproblem results.
- Tabulation: Bottom-up approach using an iterative table to store subproblem results.

DP Algorithm

Algorithm Steps

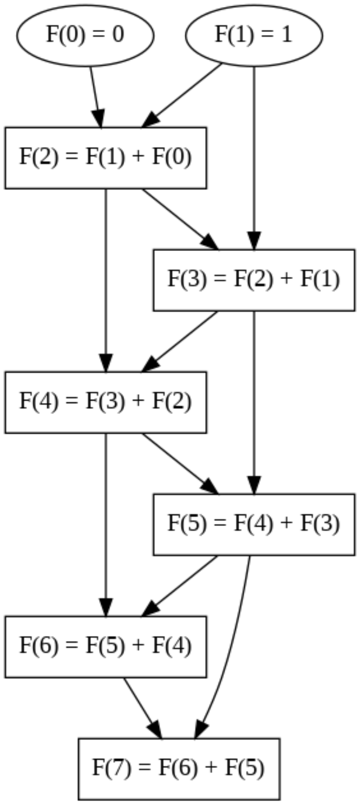
1. **Define the State:**
   - Determine how to represent subproblems as states.
2. **Identify the Recurrence Relation:**
   1. Derive how the current state depends on previous states.
3. **Determine the Best Cases:**
   - Specify the initial conditions for the smallest subproblems.
4. **Choose the Approach:**
   - Memoization vs. Tabulation

Mathematically

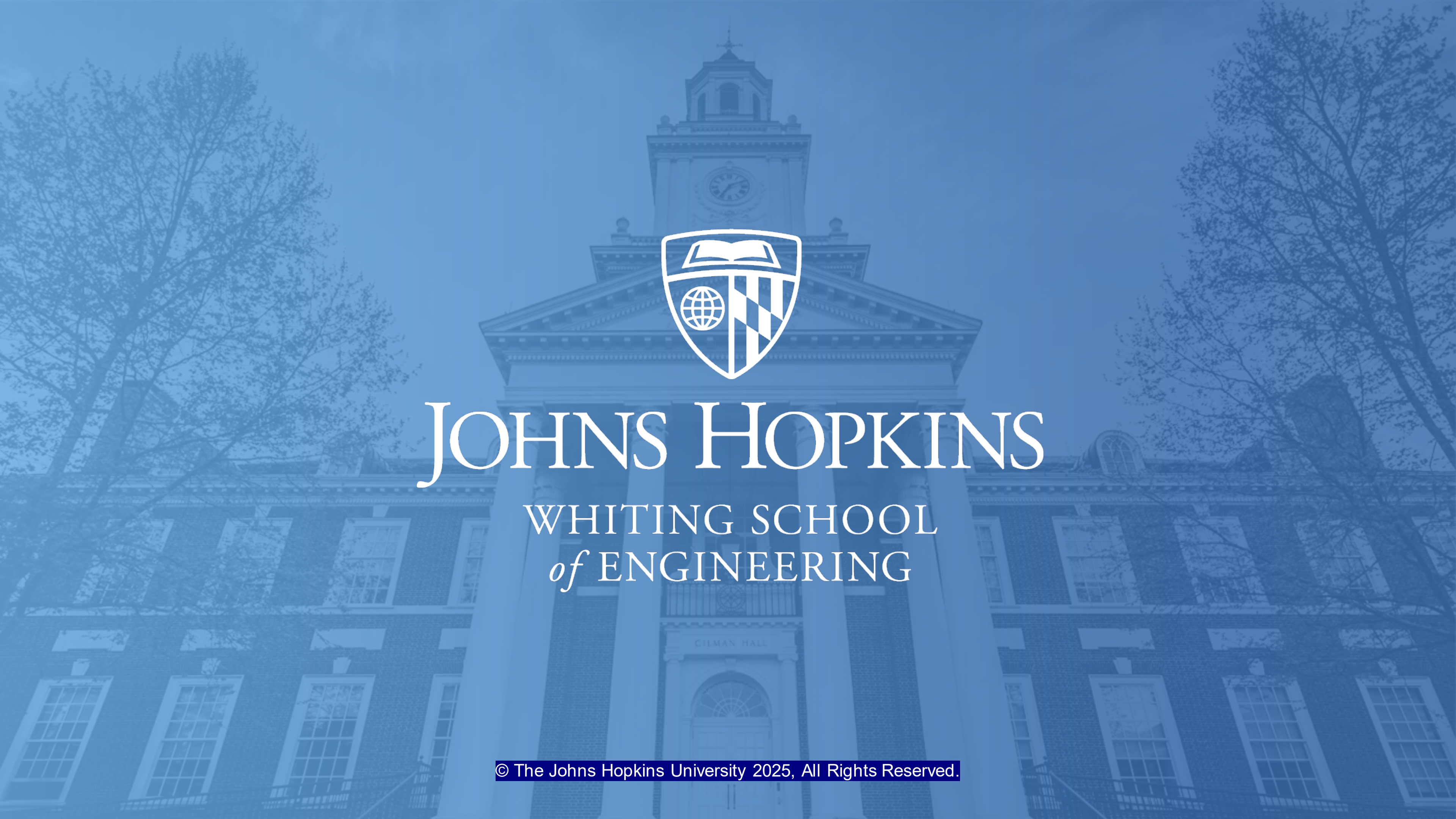
1. **Define the State:**
   - $dp[i]$ represents the solution to the problem up to index i.
2. **Identify the Recurrence Relation:**
   - $dp[i] = dp[i-1] + dp[i-2]$
3. **Determine the Best Cases:**
   - $dp[0] = 0, dp[1] = 1$
4. **Choose the Approach:**
   - Recursion/caching vs. Iterative bottom-up approach.

DP Algorithm: Fibonacci Sequence

Each cell represents a state F(n)



Each cell is computed using previously solved subproblems F(n−1) and F(n−2)


JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING