

# Concepts and Mental Model

---

DP is about solving problem with overlapping subproblems.

## The SRTBOT Framework (For Logically Rigorous Algorithm Design)

### 1. **Subproblem** definition

- Describe the meaning of subproblem in **words**, in terms of input parameters
- Subproblems usually lie in **subsets of input**
  - e.g. prefix, suffix, running (continuous) substring of a sequence
- Subproblems usually record **partial state** (of the entire solution):
  - could add subproblems by incrementing some auxiliary variables.

### 2. **Relate** subproblem solutions recursively

- look for recursive relation between subproblems
- DP recurrence must satisfy:
  1. define **only one** state at a time
  2. decisions must be made by comparing **complete subproblem solutions** instead of local costs.

### 3. **Topological order** on subproblems

- For arguing relation is *acyclic* and *subproblems form a DAG*

### 4. **Base** cases of relation

- Solution to ultra simple cases (for all reachable subproblems) when recursion (or dependency on subproblems) is not needed

### 5. **Original** problem solution via subproblem(s)

- Show how the original problem could be solved by reusing solutions to subproblem(s)

### 6. **Time** and space analysis

- Source: [MIT 6.006 Introduction to Algorithms: DP Lecture Notes](#)

## Greg Hogg's Approach (More Interview-Style, Time Limited Approach)

- This is essentially compressed SRTBOT

recursive backtrack -> top-down memoization -> bottom-up tabulation -> bottom-up without table (or only partial table)

- source: <https://www.youtube.com/shorts/uUjFL0C-vY0?feature=share>

## How to Verify Recursion

---

Pause and ask these guiding questions (assume  $f(x)$  is the recursive function):

1. What exactly does  $f(x)$  mean?
2. Do my base cases match reality?
3. What is the last decision?
4. Are the subproblems disjoint and complete?

5. *Does this match small hand-calculated cases?*

## Cache/Memoization Notes:

---

1. Cache/Memo only remembers **past computation**, which are computed by recursion.
2. Memoization stores answers to states, not transitions between states.