3-1 Dynamic Programming
(Problems and "Theory")

Hengfeng Wei

hfwei@nju.edu.cn

September 10, 2018

Taolu

Steps for Applying DP:

Steps for Applying DP:

(I) Define subproblems

(II) Set the goal

(III) Identify the recurrence
- larger subproblem $\leftarrow$ # smaller subproblems
- init. conditions

(I) Define subproblems

(II) Set the goal

(III) Identify the recurrence
  - larger subproblem $\leftarrow$ # smaller subproblems
  - init. conditions

(IV) Write pseudo-code: filling in "tables" in some order

(V) Analyze the time complexity

(VI) Extract the optimal solution (optionally)

Steps for Applying DP:

(I) Define subproblems

(II) Set the goal

(III) Identify the recurrence
  - larger subproblem $\leftarrow$ # smaller subproblems
  - init. conditions

(IV) Write pseudo-code: filling in "tables" in some order

(V) Analyze the time complexity

(VI) Extract the optimal solution (optionally)

Steps for Applying DP:

(I) Define subproblems

(II) Set the goal

(III) Identify the recurrence
- larger subproblem $\leftarrow$ # smaller subproblems
- init. conditions

(IV) Write pseudo-code: filling in "tables" in some order

(V) Analyze the time complexity

(VI) Extract the optimal solution (optionally)

1D Subproblems

Input: $x_1, x_2, \ldots, x_n$ (array, sequence, string)

Subproblems: $x_1, x_2, \ldots, x_i$ (prefix/suffix)

\#: $\Theta(n)$

Examples:
- Rod cutting
- Maximum-sum subarray
- Longest increasing subsequence
- Text justification (LaTeX)

2D Subproblems

(I) Input: $x_1, x_2, \ldots, x_m;$ $y_1, y_2, \ldots, y_n$

Subproblems: $x_1, x_2, \ldots, x_i;$ $y_1, y_2, \ldots, y_j$

$\#$: $\Theta(mn)$

Examples: Edit distance, Longest common subsequence

2D Subproblems

(I) Input: $x_1, x_2, \ldots, x_m; \quad y_1, y_2, \ldots, y_n$

Subproblems: $x_1, x_2, \ldots, x_i; \quad y_1, y_2, \ldots, y_j$

$\#$: $\Theta(mn)$

Examples: Edit distance, Longest common subsequence

(II) Input: $x_1, x_2, \ldots, x_n$

Subproblems: $x_i, \ldots, x_j$

$\#$: $\Theta(n^2)$

Examples: Matrix chain multiplication, Optimal BST

3D Subproblems

▶ Floyd-Warshall algorithm

$$\mathsf{d}(i, j, k) = \min\Big(\mathsf{d}(i, j, k-1), \mathsf{d}(i, k, k-1) + \mathsf{d}(k, j, k-1)\Big)$$

DP on Graphs

(I) On rooted tree

Subproblems: rooted subtrees

(II) On DAG

Subproblems: nodes after/before in the topo. order

DP on Graphs

(I) On rooted tree

Subproblems: rooted subtrees

(II) On DAG

Subproblems: nodes after/before in the topo. order

Knapsack Problem

Subset sum problem, Change-making problem
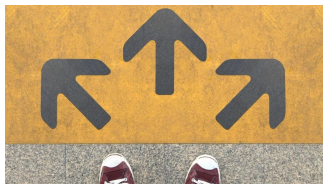
And Others . . .

And Others . . .
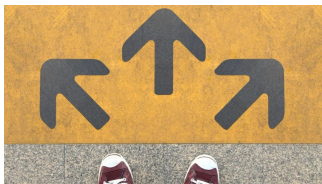
How to identify the recurrence?

How to identify the recurrence?

GUESS

Make Choices by asking yourself the right question

Make Choices by asking yourself the right question

(I) Binary choice
- whether . . .

(II) Multi-way choices
- where to . . .
- which one . . .