Dynamic Programming

Hengfeng Wei

hfwei@nju.edu.cn

June 11 – June 19, 2017

Dynamic Programming

What is DP?

DP $\approx$ " brute force"

DP $\approx$ "smart scheduling of subproblems"

DP $\approx$ "shortest/longest paths in some DAG"

What is DP?

DP $\approx$ "smarter brute force"

DP $\approx$ "smart scheduling of subproblems"

DP $\approx$ "shortest/longest paths in some DAG"

What is not DP?

Programming = Planning

What is not DP?

$$\text{Programming} = \text{Planning}$$

$$\text{Programming} \neq \text{Coding}$$

(Richard Bellman, 1940s)

Steps for applying DP

1. Define subproblems
   - # of subproblems
2. Set the goal
3. Define the recurrence
   - larger subproblem $\leftarrow$ # smaller subproblems
   - init. conditions
4. Write pseudo-code: fill "table" in topo. order
5. Analyze Time/Space complexity
6. Extract the optimal sulution

Common subproblems in DP

1D subproblems:

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

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

#: $\Theta(n)$

Common subproblems in DP

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: Fib, Maximum-sum subarray, Longest increasing subsequence, Highway restaurants, Text justification

Common subproblems in DP

2D subproblems:

1. Input: $x_1, x_2, \ldots, x_m$; $y_1, y_2, \ldots, y_n$

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

   #: $\Theta(mn)$

Common subproblems in DP

2D subproblems:

1. 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

Common subproblems in DP

2D subproblems:

1. 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

2. Input: $x_1, x_2, \ldots, x_n$

   Subproblems: $x_i, \ldots, x_j$

   #: $\Theta(n^2)$

Common subproblems in DP

2D subproblems:

1. 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

2. Input: $x_1, x_2, \ldots, x_n$

   Subproblems: $x_i, \ldots, x_j$

   #: $\Theta(n^2)$

   Examples: Matrix chain multiplication, Optimal BST

Common subproblems in DP

3D subproblems:

- Floyd-Warshall algorithm

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

Common subproblems in DP

3D subproblems:

- Floyd-Warshall algorithm

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

DP on graphs:

1. On rooted tree

   Subproblems: rooted subtrees

2. On DAG

   Subproblems: nodes after/before in the topo. order

Common subproblems in DP

3D subproblems:

- Floyd-Warshall algorithm

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

DP on graphs:

1. On rooted tree

   Subproblems: rooted subtrees

2. On DAG

   Subproblems: nodes after/before in the topo. order

Knapsack problem:

- Subset sum problem, change-making problem

Common subproblems in DP

And Others . . .

Dynamic Programming

Dynamic Programming

Dynamic Programming

Dynamic Programming


THANK YOU