3-1 Dynamic Programming

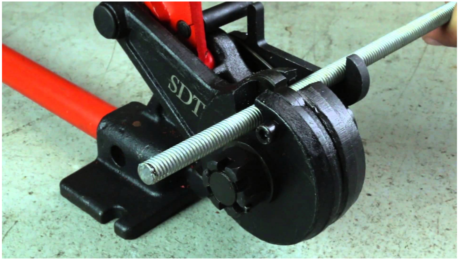
(Part I: Examples)

Hengfeng Wei

hfwei@nju.edu.cn

September 10, 2018

Rod Cutting



Rod Cutting Problem

Rod of length n

Rod Cutting Problem

Rod of length n

length i	1	2	3	4	5	$\cdots$
price p_i	1	5	8	9	10	$\cdots$

Rod Cutting Problem

Rod of length n

length i	1	2	3	4	5	$\cdots$
price p_i	1	5	8	9	10	$\cdots$

$$n = i_1 + i_2 + \cdots + i_k$$

$$r_n = p_{i_1} + p_{i_2} + \cdots + p_{i_k}$$

Subproblem: $R(i)$: $\max$ revenue obtained from cutting a rod of length i

Goal: $R(n)$

Subproblem: $R(i)$: $\max$ revenue obtained from cutting a rod of length i

Goal: $R(n)$

Make Choice: Where is the *first* cut?

Recurrence:

$$R(i) = \max_{1 \le j \le i} \Big(p_j + R(i-j) \Big)$$

Subproblem: $R(i)$: $\max$ revenue obtained from cutting a rod of length i

Goal: $R(n)$

Make Choice: Where is the *first* cut?

Recurrence:

$$R(i) = \max_{1 \le j \le i} \Big(p_j + R(i - j) \Big)$$

Init:

$$R(0) = 0$$

Subproblem: $R(i)$: $\max$ revenue obtained from cutting a rod of length i

Goal: $R(n)$

Make Choice: Where is the *first* cut?

Recurrence:

$$R(i) = \max_{1 \le j \le i} \Big(p_j + R(i-j) \Big)$$

Init:

$$R(0) = 0$$

Time:

$$O(n^2) = \Theta(n) \cdot O(n)$$

Rod Cutting Problem (Problem 15.1-3)

Each cut incurs a fixed cost of c.

Rod Cutting Problem (Problem 15.1-3)

Each cut incurs a fixed cost of c.

$$R(i) = \max_{1 \le j \le i} \Big(p_j - c + R(i - j) \Big)$$

Rod Cutting Problem (Additional)

$\leq m$ cuts

Rod Cutting Problem (Additional)

$\leq m$ cuts

Subproblem: $R(i, k)$: $\max$ revenue obtained from cutting a rod of length i using $\leq k$ cuts

Goal: $R(n, m)$

Rod Cutting Problem (Additional)

$\leq m$ cuts

Subproblem: $R(i, k) : \max$ revenue obtained from cutting a rod of length i using $\leq k$ cuts

Goal: $R(n, m)$

Make Choice: Where is the *first* cut?

Rod Cutting Problem (Additional)

$\leq m$ cuts

Subproblem: $R(i,k) : \max$ revenue obtained from cutting a rod of length i using $\leq k$ cuts

Goal: $R(n,m)$

Make Choice: Where is the *first* cut?

Recurrence:

$$R(i,k) = \max_{i+1 \leq j \leq i} \Big(p_j + R(i-j, k-1) \Big)$$

Rod Cutting Problem (Additional)

$\leq m$ cuts

Subproblem: $R(i, k)$: $\max$ revenue obtained from cutting a rod of length i using $\leq k$ cuts

Goal: $R(n, m)$

Make Choice: Where is the *first* cut?

Recurrence:

$$R(i, k) = \max_{i+1 \leq j \leq i} \Big(p_j + R(i - j, k - 1) \Big)$$

Init:

$$R(0, k) = 0 \quad R(i, 0) = p_i$$

Rod Cutting Problem (Additional)

$\leq m$ cuts

Subproblem: $R(i,k)$: $\max$ revenue obtained from cutting a rod of length i using $\leq k$ cuts

Goal: $R(n,m)$

Make Choice: Where is the *first* cut?

Recurrence:

$$R(i,k) = \max_{i+1 \leq j \leq i} \Big(p_j + R(i-j, k-1)\Big)$$

Init:

$$R(0,k) = 0 \quad R(i,0) = p_i$$

Time:

$$O(n^2 m) = \Theta(nm) \cdot O(n)$$

Matrix-chain Multiplication

Subproblem: $m[i, j]$: min cost to compute the matrix $A_{i \ldots j}$

Goal: $m[1, n]$

Subproblem: $m[i, j]$: min cost to compute the matrix $A_{i \ldots j}$

Goal: $m[1, n]$

Make Choice: Where is the *last* parentheses?

Recurrence:

$$m[i, j] = \min_{i \le k < j} \Big(m[i, k] + m[k+1, j] + p_{i-1} p_k p_j \Big)$$

Subproblem: $m[i,j]$: min cost to compute the matrix $A_{i\ldots j}$

Goal: $m[1,n]$

Make Choice: Where is the *last* parentheses?

Recurrence:

$$m[i,j] = \min_{i \le k < j} \Big(m[i,k] + m[k+1,j] + p_{i-1}p_k p_j \Big)$$

Init:

$$m[i,i] = 0$$

Subproblem: $m[i,j]$: min cost to compute the matrix $A_{i \dots j}$

Goal: $m[1,n]$

Make Choice: Where is the *last* parentheses?

Recurrence:

$$m[i,j] = \min_{i \le k < j} \Big(m[i,k] + m[k+1,j] + p_{i-1} p_k p_j \Big)$$

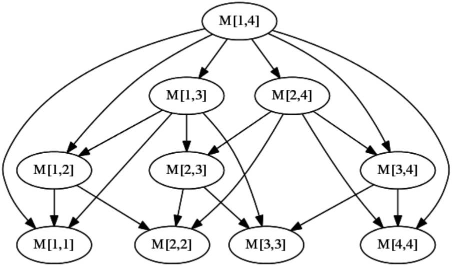
Init:

$$m[i,i] = 0$$

Time:

$$O(n^3) = \Theta(n^2) \cdot O(n)$$

Subproblem Graph for Matrix-chain Multiplication (Problem 15.2-4)



Triangulation

$T(i,j)$: Cost of triangulating from v_i to v_j

$$T(i,j) = \min_{i<k<j} \Big(T[i,k] + T[k,j] + d_{ik} + d_{kj} \Big)$$


THANK YOU