

## 动态规划算法

胡船长

初航我带你,远航靠自己



## 动态规划经典算法讲解

大约用时: (90 mins)

下一部分: 经典面试刷题环节

### 数学归纳法



Step1:验证 k<sub>0</sub>成立

Step2: 证明如果  $k_i$  成立,那么  $k_{i+1}$  也成立

Step3: 联合 Step1 与 Step2,证明由 k<sub>0</sub>->k<sub>n</sub>成立 4

### 数 学 归 纳 法



Step1: 验证 k<sub>0</sub> 成立

Step2:证明如果  $k_i$  成立,那么  $k_{i+1}$  也成立

Step3: 联合 Step1 与 Step2,证明由 k<sub>0</sub>->k<sub>n</sub>成立

dpZi1ZjZ

思考-1:如果由下向上走的话,很容易获得最后一行站在<u>每个点</u>上能够获得的最大值,由此得到倒数第二行的<u>每个点</u>的最大值,依次递推。

思考-2:如果由上向下走的话,很容易获得第一行站在<u>每个点</u>上能够获得的最大值,由此得到第二行的<u>每个点</u>的最大值,依次递推。

## 状态转移方程中的重点 海贼宝



状态:一个数学符号,外加一个语义描述

决策:从所有可能产生最优解的状态中,选择一个最大值。

阶段:本阶段只依赖于上一个阶段(这个概念,意会即可)

### 概念对应-数字三角形



1、确定动归状态

例如: f(i,j) 代表从底边走到(i,j) 点所能获得的最大值

2、确定状态转移方程

例如: 
$$f(i,j) = max \left\{ f(i+1,j) \atop f(i+1,j+1) \right\} + val(i,j)$$

- 3、正确性证明: 求助于数学归纳法
- 4、程序实现

### 递推问题的求解方向



#### 1、我从哪里来

目前为止,见到的动态规划问题,求解过程,均是这种。

#### 2、我到哪里去

5 5 5 5 5 5

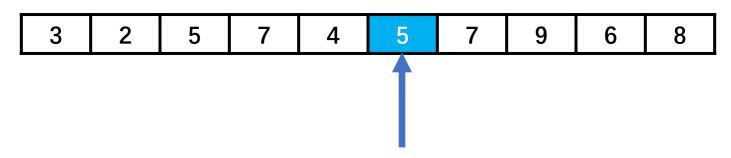
## 练习题1:最长上升子序列 海贼宝藏

【编写程序】有一个数字序列,求其中最长上升子序列的长度

3	2	5	7	4	5	7	9	6	8
---	---	---	---	---	---	---	---	---	---

## 练习题1:最长上升子序列 海贼宝藏

【编写程序】有一个数字序列,求其中最长上升子序列的长度



考虑以立位作为结尾的最长上升子序列长度

## 练习题1:最长上升子序列

#### 1、确定动归状态

例如: dp(i) 代表以 i 位为结尾的上升子序列最长长度

2、确定状态转移方程

3、时间复杂度分析

求解 dp(i) 需要遍历 i - 1 次,所以总体时间复杂度为  $O(n^2)$ 

### 



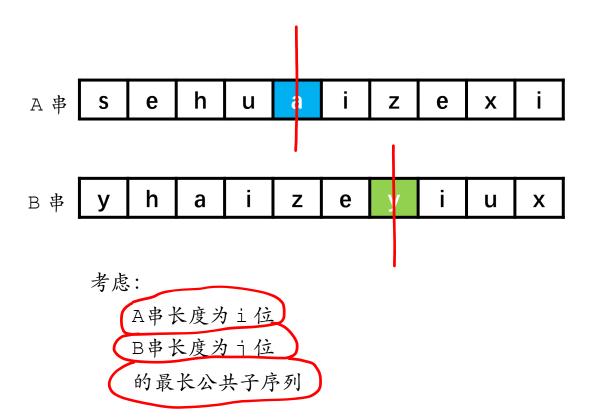
【编写程序】给出两个字符串,求其两个的最长公共子序列。



### 练习题2: 最长公共子序列



【编写程序】给出两个字符串,求其两个的最长公共子序列。



### 练习题2: 最长公共子序列 海贼宝



#### 1、确定动归状态

\_dp(i,j) 代表

A串长度为 i 位, B串长度为 j 位的最长公共子序列长度

#### 2、确定状态转移方程

$$dp(i,j) = max \begin{cases} dp(i,j-1) \\ dp(i-1,j) \\ dp(i-1,j-1) + (A[i] == B[j]) \end{cases}$$

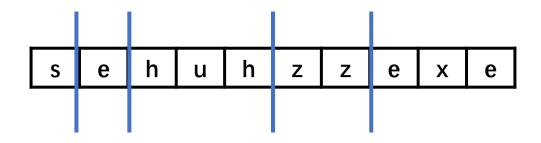
#### 3、分析转移方程意义

第一项和第二项可以理解成为是A[i]和B[j]不相等的情况 第三项当A[i]与B[i]相等时,才有意义,不过不影响结果

## 练习题3:切割回文



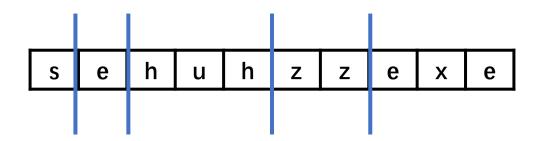
【编写程序】给出一个字符串S,问对字符串S最少切几刀,使得分成的每一部分都是一个回文串(注意:单一字符,是回文串)。



## 练习题3:切割回文



【编写程序】给出一个字符串S,问对字符串S最少切几刀,使得分成的每一部分都是一个回文串(注意:单一字符,是回文串)。



考虑:字符串短的时候,是否好解决?

## 练习题3:切割回文



#### 1、确定动归状态

dp(i,j) 代表从i到j最少切多少刀

#### 2、确定状态转移方程

$$dp(i,j) = min \begin{cases} dp(i,k) + dp(k+1,j) + 1 | k \in [i,j) \\ 0 | S[i] = S[j] \perp dp(i+1,j-1) = 0 \end{cases}$$

#### 3、分析转移方式

大区间的结果由其包含的小区间确定,转移方式就应该是先确定 小区间的结果,逐渐增大区间范围。

很容易确定所有 dp(i, i) 的值

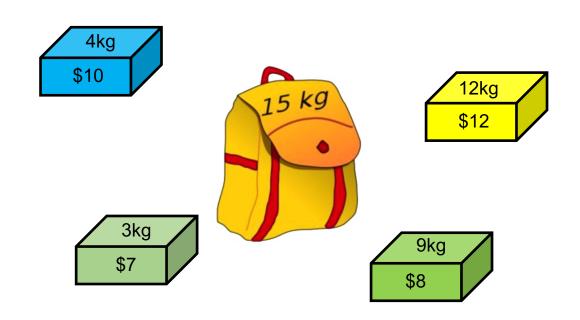
第二步确定所有 dp(i, i + 1) 的值

第三步确定所有 dp(i, i + 2) 的值…

### 练习题4: 0/1背包



【编写程序】给有一个能承重 V 的背包,和 n 件物品,我们用重量和价值的二元组来表示一个物品,第 i 件物品表示为  $(v_i, w_i)$  ,问在背包不超重的情况下,得到物品的最大价值是多少?



### 练习题4: 0/1背包



#### 1、确定动归状态

dp(i,j)代表前i件物品,背包承重为j时所获得的最大价值

#### 2、确定状态转移方程

$$dp(i,j) = max \begin{cases} dp[i-1][j] & | 不选第 i 件物品 \\ dp[i-1][j-v[i]] + w[i] & | 选第 i 件物品 \end{cases}$$

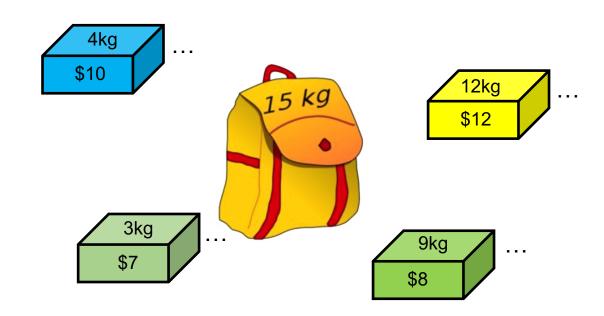
#### 3、填表格

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
4,10																
3,7																
9,8																
12 <b>,</b> 1																

## 练习题5:完全背包



【编写程序】给有一个能承重 V 的背包,和 n <u>种</u>物品,<u>每种数量任意</u> <u>多</u>,我们用重量和价值的二元组来表示一个物品,第 i 种物品表示为  $(V_i, W_i)$ ,问在背包不超重的情况下,得到物品的最大价值是多少?



## 练习题5:完全背



#### 1、确定动归状态

dp(i,j)代表前i件物品,背包承重为j时所获得的最大价值

#### 2、确定状态转移方程

$$dp(i,j) = max \begin{cases} dp[i-1][j] & | 不选第 i 件物品 \\ dp[i][j-v[i]] + w[i] & | 选第 i 件物品 \end{cases}$$

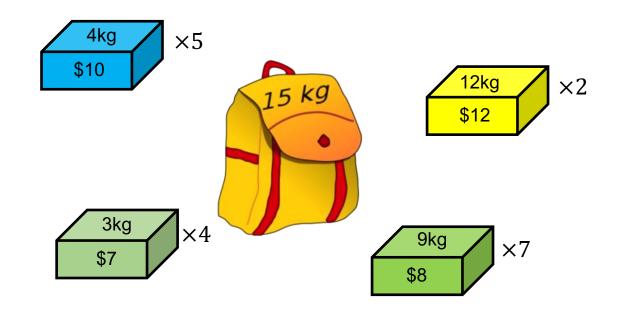
#### 3、填表格

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
4,10																
3 <b>,</b> 7																
9,8																
12,1																

## 练习题6:多重背



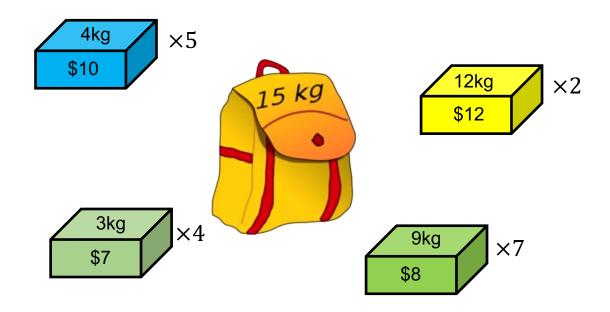
【编写程序】给有一个能承重 V 的背包,和 n <u>种</u>物品,<u>每种物品的数</u> <u>量有限多</u>,我们用重量、价值和数量的三元组来表示一种物品,第 i 种物品表示为  $(v_i, w_i, s_i)$ ,问在背包不超重的情况下,得到物品的最大价值是多少?



## 练习题6:多重背



【思考】如下图,可否当成是包含18件物品的0/1背包做?



### 练习题6:多重背



### 包

#### 1、确定动归状态

dp(i,j)代表前i件物品,背包承重为j时所获得的最大价值

#### 2、确定状态转移方程

$$dp(i,j) = max\{dp[i-1][j-k*v[i]] + k*w[i]\}$$
  
第 i 种物品选择 k 件,  $k \in [0,s_i]$ 

#### 3、填表格

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
4,10																
3,7																
9,8																
12,12								-								

注: (4, 10, 5)、(3, 7, 4)、(9, 8, 7)、(12, 12, 2)

### 动归问题总结



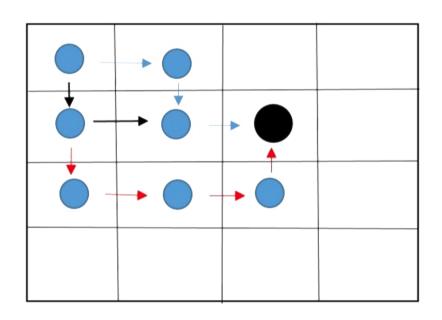
- 1、动态规划是一类特殊的递推问题 满足递推问题求解的三步走原则
- 2、最优化问题 最大、最小、最长…
- 3、最优子结构 规模为 n 的最优解,取决于规模为 n-1的最优解
- 4、无后效性 之前的决策对后续决策无影响

### 动归问题总结



#### 无后效性举例-1:

如下图,有一个二维表格,从左上角的点出发,每次能往右或者 往下走,每个格子上有一个数字,请问,从左上角走到右下角能够获得的 最大值是多少?

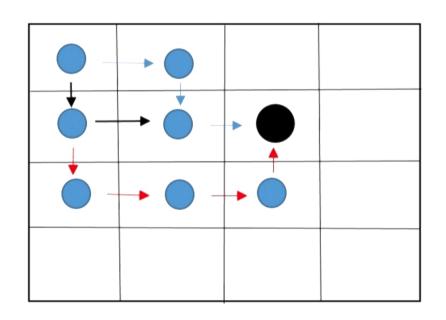


### 动归问题总结



#### 无后效性举例-2:

如下图,有一个二维表格,从左上角的点出发,每次能往<u>上下左</u> <u>右四个方向</u>走,每个格子上有一个数字,请问,<u>在不走重复格子</u>的情况下, 从左上角走到右下角能够获得的最大值是多少?







Luogu	P1507	Luogu	P1233
Luogu	P1131	Luogu	P2430
Luogu	P1481	Luogu	P2426
Luogu	P1509	Luogu	P1140
Luogu	P1436	Luogu	P1220
Luogu	P1624	Luogu	P1435
Luogu	P1103	Luogu	P1156
Luogu	P1272	Luogu	P1651
Luogu	P2018	Luogu	P1681
Luogu	P2719	Luogu	P1280



## 经典面试题刷题专项环节

大约用时: (120 mins)

下一部分: 浪尖人才事业部

### 问题板书





# 每天都想干翻这个世界到头来,被世界干的服服帖帖

大家晚安