动态规划

Ben

课程说明

- 前导技能
  - 递归，基本的暴力搜索（必会）
- 动态规划 (Dynamic Programming)
  - 名字并无多大意义
- 目标
  - 分析 -> Coding -> AC
- 课程开始
  - Leetcode 198

定义

- 本质：递归
- 原问题（N）->子问题（N-1）->原问题（N）
- 最优子结构
  - 子问题最优决策可导出原问题最优决策
  - 无后效性
- 重叠子问题
  - 去冗余
  - 空间换时间（注意分析时空复杂度）

问题共性

- 套路
  - 最优，最大，最小，最长，计数
- 离散问题
  - 容易设计状态（整数01背包问题）
- 最优子结构
  - N-1可以推导出N

基本步骤

- 四个步骤
  - 设计暴力算法，找到冗余
  - 设计并存储状态（一维，二维，三维数组，甚至用Map）
  - 递归式（状态转移方程）
  - 自底向上计算最优解（编程方式）

实例

- 斐波那契数列
  - 暴力递归
  - F(n)表示斐波那契第n个
  - F(n) = F(n-1) + F(n-2), if n>=2, otherwise F(n) = 1
  - for i <- 2 to n
- N！
  - 暴力递归
  - F(n)表示n!的值
  - F(n) = F(n-1) * n, if n>=1, otherwise F(n) = 1
  - for i <- 2 to n

小兵向前冲

- N*M的棋盘上，小兵要从左下角走到右上角，只能向上或者向右走，问有多少种走法
- 套路：计数问题
  - 暴力搜索（回溯法）
  - F(n,m)表示棋盘大小为n*m时走法数量
  - F(n,m) = F(n-1,m) + F(n,m-1) if n*m > 0, otherwise F(n,m) = 1
  - for i <- 1 to n
    - for j <- 1 to m

小兵向前冲

- 空间复杂度？
- 时间复杂度？
- 组合数递推公式C(n,m) ？
- 如果某些格子禁止小兵进入？
- 小兵一次某方向上可以走一步或者两步？

01背包问题

- 小偷有一个容量为W的背包，有n件物品，第i个物品价值vi，且重wi
- 目标: 找到xi使得对于所有的xi = {0, 1}
- sum(wi*xi) <= W, 并且 sum(xi*vi)最大
- 套路：最大
  - 暴力回溯法怎么写？
  - F(i, W)表示前i件物品体积为w，最大价值
  - F(i, W) = max{F(i-1, W), F(i-1, W-wi) + vi} //后一项当W < wi时为0
  - for i <- 1 to W
    - for j <- 1 to n

01背包问题

- 空间复杂度？
- 时间复杂度？
- 滚动数组
- 如果W很大怎么办？
- 如果相关值是实数怎么办？
- 类似题目Leetcode 322

最长公共子序列

- 一个给定序列的子序列是在该序列中删去若干元素后得到的序列
- 给定两个序列X和Y，当另一序列Z既是X的子序列又是Y的子序列时，称Z是序列X和Y的公共子序列
- 最长公共子序列
- X = (A, B, C, B, D, A, B)　Y = (B, D, C, A, B, A)
- (B, C, B, A)　(B, D, A, B)

最长公共子序列

- 如何暴力回溯递归？
- 状态表示F(i, j)
- F(i, j) = max{F(i - 1, j), F(i, j - 1), F(i -1, j-1) + 1} //后一项当X[i] != Y[i]时为零
- for i <- 1 to N
    - for j <- 1 to M
- 类似题目Leetcode 72

旅行商问题

- 一个商人要不重复的访问N个城市，允许从任意城市出发，在任意城市结束。现已知任意两个城市之间的道路长度
- 求城市的访问序列，使得商人走过的路程最短
- 例：N=4，访问序列3，4，1，2
- NP问题，最优解无多项式时间算法
- 时间复杂度？空间复杂度？
- 状态压缩
    - 时间复杂度
    - 空间复杂度

总结

- 动态规划算法用到的题目存在很多套路
- 滚动数组，状态压缩，升维，单调性，四边形不等式（高级套路）
- 先学套路，跳出套路
- 本质：先暴力，找冗余，去冗余

- 微博： BenLin_BLY
- Q&A