# 动态规划选讲

清华大学 徐明宽

# 目录

- 概述
- 序列DP
- 数位DP
- 区间DP
- 树形DP
- 状压DP
- 期望DP
- DP套DP
- 简单的动态DP

# 概述

- DP：Dynamic Programming
- 将问题用“状态”表示，存在一个方法可以依次求解每个状态的值

- 例：求Fibonacci数列的第n项（$n \le 100000$）
- 例：解线性方程组不能使用动态规划算法
- 例：堆优化Dijkstra、SPFA、Floyd等最短路算法也是DP；特别地，求DAG最短路/最长路更是DP了

# 概述

- “动态规划四要素”：
  - 状态定义
  - 转移
  - 初值
  - 所求结果
- 其中状态定义最为关键
- 总时间复杂度 = 状态复杂度 * 转移复杂度

# 概述

- 如何定义状态？
- 如果原题有状态，直接照搬定义
- 如果原题有序列，直接基于序列或区间定义状态
- 如果原题有二维数组，直接基于二维数组定义二维状态
  - 例：求二维区间和（无修改）
- 如果都不适用，则需要寻找一种定义方式，可以依次求解每个状态的值
- 如果定义状态以后发现转移是一个区间，考虑前缀和优化等

# 序列DP

- 求最长上升子序列长度。
- $n \le 1e5$

# 序列DP

- f[i]表示长度为i的最长上升子序列的最后一项的最小值
- O(n log n)
- DP时隐藏了“原序列前缀长度”这一维

# 序列DP

- 数轴上有n个村庄，在第i个村庄建立监测站的代价是ai，每个监测站可以覆盖周围len长度。
- 现在要建立若干个监测站，覆盖所有村庄，求最小代价。
- $n \leq 1e6$，坐标范围 $\leq 1e9$
- 输入坐标单调递增

# 序列DP

- f[i]表示覆盖前i个村庄且在第i个村庄建立监测站的最小代价
- （f[i]覆盖到了第i个村庄往后len的位置）
- 单调队列转移
- O(n)

# 数位DP

- hdu 3652
- 求不超过n的包含子串“13”且能被13整除的数的个数。
- $n \le 10^9$，但可以做到$10^{18}$甚至$10^{1000}$

# 数位DP

- 不超过n的限制：前若干位是否与n相同
- 包含子串“13”的限制：前若干位已包含/未包含但末位是1/未包含且末位不是1
- 被13整除：前若干位模13的余数
- $2 * 3 * 13 * \log_{10}(n)$个状态

# 区间DP

- 合并石子问题
- 有n堆石子排成一列，第i堆有ai个，每次可以将相邻的两堆合并，代价为这两堆石子个数之和。
- 求合并成一堆的最小代价。
- $n \leq 300$

# 区间DP

- $f[i][j]$表示将区间$[i, j]$的石子合并到一堆的最小代价
- 状态复杂度$O(n^2)$，转移复杂度$O(n)$

- 可以用四边形不等式优化到$O(n^2)$

# 区间DP

- 一棵二叉查找树，每个点有权值，满足每条边两端的点权不互质
- 现给出一个严格递增序列，问是否可能是一棵这样的树的点权中序遍历
- $n \le 700$
- 来源：Codeforces 1025D

- 提示：先判断哪些数可以连边，将“不互质”的条件转化成一个可能连边的邻接表

# 区间DP

- f[i][j][k]表示区间[i, j]能否以k为根构造出一棵满足条件的树
- 状态复杂度$O(n^3)$，转移复杂度$O(n)$

- f[i][j][0/1]表示区间[i, j]能否以 i – 1 / j + 1 为根构造出一棵满足条件的树
- 状态复杂度$O(n^2)$，转移复杂度$O(n)$

- bitset优化：f[i][*][0], f[i][*][1], f[*][j][0], f[*][j][1], e[i][*]（邻接表）

# 练习

- TC SRM 552 div1 medium
- 给定一块n * m的田地，每个格子里只可能是百合花（L）、牵牛花（P）和空地（.）之一
- 你可以圈出两块非空、不交的矩形区域，满足“L”和“P”的数量之差的绝对值不超过K，求花的数量的最大值
- $n, m \le 30$，$K \le n * m$
- （其实可以做到 $n, m \le 70$）

# 练习

- 把“L”看成+1，“P”看成-1，“.”看成0
- f[i][k]表示前i列选出一个元素和为k的矩形包含的非零元素数量最大值
- $O(n^2m^2)$
- 如何计算答案？

# 树形DP

- 求一棵树的最大独立集。
- $n \le 1e6$

# 树形DP

- f[i][0/1]表示i这棵子树在 不选/选 i这个点时的最大独立集。
- 选i则不能选i的所有儿子，不选i则对i的所有儿子都是可选可不选

# 树形DP

- 给定一棵树，每个点有权值
- 你可以选择若干条互不相交的链，每选择一条需要支付B的代价，并获得这条链上所有点的权值之和的收益
- $n \le 1e5$
- 权值可以为负

# 树形DP

- f[i][0/1]表示i这个子树的最大收益，第二维为1表示有一条链的端点为i。

# 练习

- Codeforces 815C
- 有n个商品，第i个商品的价格为$c_i$元，但用优惠券以后可以便宜$d_i$元（$d_i < c_i$）（不能用一个优惠券但不买该商品）
- 但是对于任意$i \ge 2$，如果想在第i个商品用优惠券，就必须先在第$x_i$个商品使用优惠券（$x_i < i$）
- 现在你有b元，问最多能买多少种不同的商品
- $n \le 5000$，$b \le 1e9$

# 状压DP

- 用“L”形去覆盖一个n * m的矩形网格，求方案数模$10^9+7$的余数
- n ≤ 4，m ≤ 1e15
- 或：n ≤ 5，m ≤ 1e4，但网格里有些地方被挖掉了
- 例：

```
11166677AAADDD
1333467999ADEE
234445798BBBCE
222555888BCCCE
```

# 状压DP

- f[i][S]表示前i – 2列已填满，第i – 1列和第i列的状态为S的方案数

- 对于m巨大的情况，使用矩阵快速幂优化

# 期望DP

- 有n个格子，每次随机涂一个，求m次以后期望涂了多少个。

# 期望DP

- f[i]表示涂了i次以后的答案。

# 期望DP

- 有n个格子，每次随机涂一个，求涂满m个格子期望要多少次。

# 期望DP

- f[i]表示当前已经涂了i个格子，期望还需要多少次才能涂满m个格子。

# 期望DP

- 给定若干个小写字母组成的单词，现在每次随机打一个字母，问期望多少次才能打出任意一个给定的单词
- （即打若干个字母以后，当前打的字符串的一个后缀是一个给定的单词）

# 期望DP

- 建立AC自动机，f[i]表示当前在状态i的答案

# 期望DP

- 给定若干个小写字母组成的单词，每个单词有一个权值，现在每次随机打一个字母，每打出一个给定单词即可获得对应的权值，问打n个字母以后期望总权值是多少

# 期望DP

- 建立AC自动机，f[i][j]表示在状态j之后再打i个字母可以获得的期望权值

# 状压+期望DP

- 清华集训2017 小Y和恐怖的奴隶主
- 维护一个（可重）集合，初始有两个数{∞, m}。
- 现在进行n次操作，每次随机选一个数，把它减一：如果结果为0，把它删掉；否则如果选择的数不为∞，且集合中除了∞以外的数不超过k个，则向集合中添加一个m。
- 最后问那个∞期望被减了多少，对质数998244353取模。
- 每个测试点对于相同的m和k，询问T次（可能不同的）n。
- 极限数据 $T = 500, n \le 10^{18}, m = 3, k = 8$。

# 状压+期望DP

- 记录集合中1, 2, 3的个数，只有不超过165种可能。
- 矩阵快速幂优化DP
- $O(T\ w^3 \log n)$，60分

- 使用向量乘矩阵优化
- $O(w^3 \log n + T\ w^2 \log n)$，100分

# DP套DP

- 求长度为n的随机排列的最长上升子序列期望长度，对一个大质数取模。

- $n \leq 28$

- 例：n = 2，排列有1/2的概率为1, 2，有1/2的概率为2, 1，故最长上升子序列期望长度为1/2*(2 + 1) = 3/2。
- bzoj5161

# DP套DP

- f[i][S]表示前i个数最长上升子序列DP数组为S的概率。
- f[i][S]表示长度为i的排列的最长上升子序列DP数组为S的概率。

- 转移时，枚举长度为i+1的排列的最后一个元素，记为k，则将S中所有大于等于k的元素加一，然后更新DP数组即可。

- 状态复杂度$O(2^n)$
- 转移复杂度$O(n)$

# DP套DP

- 输入一个长度为n的排列的最长上升子序列，求原排列方案数。
- $n \le 15$
- 例：$n = 3$，输入1, 3，则原排列有2, 1, 3和1, 3, 2两种

# DP套DP

- f[S]表示原排列的一个前缀一共有哪些数，且这些数有哪些是在最长上升子序列DP数组里的

- 状态复杂度$O(3^n)$，转移复杂度$O(n)$

# 简单的动态DP

- 维护一个序列，支持：
- 查询一个区间中k个不相交的区间的和的最大值
- 修改一个点的值
- $n \leq 1e5$，$k \leq 5$

# 简单的动态DP

- 线段树+区间DP
- f[i][j][t][0/1][0/1]表示区间[i, j]中t个不相交区间的和的最大值，且i、j在不在其中某个区间内
- 每次合并$O(k^2)$
- 每次修改操作$O(k^2 \log n)$，每次询问$O(\log n)$

# 简单的动态DP

- 线段树+区间DP
- 每次询问做k次：选择最大子段和，并把这个区间取反
- 每次修改操作O(logn)，每次询问O(klogn)

# 2018 CCPC Final D

- 有一棵n个节点的树，现在Panda在节点S（算进入一次）。
- 当他进入一个节点两次以后，离开这个节点的同时，这个节点将被删除。
- 现在Panda会不停地等概率随机选择一个（没被删除的）邻居，走过去。
- Panda肯定会最后在某个节点卡住。第i个节点有个权值vi，求最后Panda卡住的节点的期望权值，对$(10^9+7)$取模。
- $T \le 10$组数据，$n \le 10^5$，$vi \le 1000$。

# 2018 CCPC Final D

- ansf[i]：i子树的答案（现在在i，i没被删，i的父亲被删了）
- 由ansf[i]，考虑从i走到儿子son，然后再也没走到i，最终卡在son子树里了：
  - f[i]：i子树的答案（现在在i，i没被删，i的父亲没被删但我们强制要求不走i向父亲的边）
- 由ansf[i]，考虑从i走到儿子son，然后紧接着走到i，再走到son，最终卡在son子树里了：
  - 如果son是个叶子，直接val[son]
  - 否则，son已经被删了，我们可以再走到son的儿子。记sumansf[i] = sum(son) ansf[son]，那么这种情况就是sumansf[son] / (deg[son] – 1)

# 2018 CCPC Final D

- 由ansf[i]，考虑从i走到儿子son，然后紧接着走到i，再走到一个别的儿子son2，最终卡在son2子树里了：
  - sum(son2 != son) ansf[son2]
  - 即sumansf[i] – ansf[son]
- 由ansf[i]，剩下的都是从i往下先走2层后来又回到i的情况。

- 记g[i]表示i的父亲没被删，从i往下走至少2层又回到i的概率。
- 记g1[i]表示i的父亲没被删，从i往下走恰好1层又回到i的概率。

# 2018 CCPC Final D

- g和g1的初值都是叶节点概率为0。
- g1可以直接算。
- g可以用g和g1算。
- 由ansf[i]，考虑从i走到儿子son（此时son的父亲没被删，符合g和g1的定义），然后要么往下走恰好1层又回到son，要么往下走至少2层又回到son，之后紧接着走回i（如果往别的地方走，就回不到i了，就和之前统计的重复了），然后由于son已经被删了只能走到另一个儿子son2：
  - sum(son2 != son) ansf[son2]
  - 即sumansf[i] – ansf[son]

# 2018 CCPC Final D

- f[i]的推法完全类似。
- 最后一遍dfs搞定，时间复杂度O(n)。

# 练习

- Codeforces 814E
- 有一个简单无向图，满足以下要求：
  - 每个点到1号点的最短路存在且唯一
  - 2到n号点到1号点的最短路单调不降
  - 第i个点的度数等于给定的$d_i$，且输入满足$d_i = 2$或3
- 求原图方案数模$10^9+7$。
- $n \le 50$

谢谢大家！