Rummikub

- 当前有N张牌，每张牌有一个编号和一个颜色，编号有1到100共计100种，颜色有r/g/b/y共4种颜色。可以对牌进行消除，消除方式有两种：
- 1）同颜色，连续3个及以上的数字，比如：9r 10r 11r
- 2）3张或以上相同数字，不同颜色的牌，比如：1r 1g 1b
- 每做一次消除，可以获得牌上所有数字之和作为得分
- 给定N，和N张互不相同的牌，求出最大可能得分
- 1<=N<=400

Rummikub

- 5
- 7g 7b 7r 8r 9r

- 7
- 23b 1y 24b 1r 93b 1b 100r

- 8
- 2y 2r 2g 2b 4g 5g 6g 7g

- 24
- 3
- 30

pick your players

- 你是一个足球队的教练，现在需要你去组织一支足球队，现在有N个球员可以挑选，每个球员有三个参数：位置（前锋、中场、后卫、守门员）、价值、价格
- 你要选取11个人，恰好有1个守门员，3～5个后卫，2～5个中场，1～3个前锋
- 你需要从中选取一个队长
- 挑出的队伍，价格为11个球员的总和，价值为11个球员的总和再多加上队长的价值
- 给定N个球员，价格限制T，求可以得到的队伍最大价值、在最大价值情况下的价格、和可以取出的不同方案数（11个球员有一个不同即视为方案数不同，队长不考虑）
- 1<= N 、T <= 1000

树形DP

- 递推关系是树形关系 而非线性关系

- 直接在树上DP

- 树的转化： 树 —> 二叉树 “左孩子 右兄弟”原则

- （树规七题）

状态压缩DP

- 状态 如何 压缩?
    - 二进制 代表0/1状态
    - 省略无用变量

思考

- 给定一个1*N的棋盘，放K个棋子，两两不相邻的方案数？ N <= 100

- 给定一个N*M的棋盘，放K个棋子，两两不相邻的方案数，四连通 / 八连通？ N*M <= 40

思考

- 给定一个N*M的棋盘，放K个棋子，两两不相邻的方案数，四连通 / 八连通?
- N <= 5 M <= 10^9

思考

- F[N][A][B][C]
- 操作了N步，每种操作分别使用了A\B\C次数

思考

- F[N][A][B][C]
- 操作了N步，每种操作分别使用了A\B\C次数

- N = A+B+C 省去一维变量
- 省去哪一位?

数位DP

- 枚举每一个数字位上的取值，同时取值的数字有一定的限制
- 常见限制：
    - 相邻不成出现XX
    - 枚举的数字不超过XX

数位DP

- 思考：
  - N位数中数位上不出现连续的262有多少个
  - [0, b]区间内回文数一共有几个

数位DP

- 数字x在十进制下表示成(AnAn-1An-2 … A2A1), F(x)记为 $A_n * 2^{n-1} + A_{n-1} * 2^{n-2} + \ldots + A_2 * 2 + A_1 * 1.$
- 求在[0, b]范围内有多少数x满足F(x) <= F(a)
- 多组数据T <= 10^4 a, b<=10^9

数位DP

- 给定两个正整数a和b，求在[a,b]中的所有整数中，每个数码(digit)各出现了多少次。

- 30% a < b <= 10^6
- 100% a < b <= 10^12

CF 55D Beautiful Number

- 找出区间[l, r]中的beautiful number个数，beautiful number的定义是这个数字可以被每一个位上的数字整除

- l < r <= 9 * 10^18

插头DP

- 基于联通性的状压DP问题
    - 不只是01描述状态，而用数字表示多种状态
    - 轮廓线
- ref：cdq《基于连通性状态压缩的动态规划问题》

动态规划优化

- 线段树优化
- 斜率优化
- 决策单调性优化
- 单调队列优化
- 四边形不等式优化

线段树优化

- 什么是线段树优化?
- 用线段树处理区间最大 / 最小值
- Sample：最长上升子序列

线段树优化

- 线段树的优化通常在状态转移中牵涉到一段区间内信息的维护，所以通常会有比较明显的提示，所以会很显然的想到用线段树（特殊情况下可以用树状数组）来维护区间的求值，将O(N)查询降低为O(logN)

单调队列优化

- 状态转移公式为 dp[i] = min / max{f[k]} + x[i]
- 这样我们对f[k]就可以使用单调队列优化
- 注意到如果k的取值是 1..i-1 那么就可以直接用一个变量记录
- 但如果k的取值是一段范围 比如说 i-m .. i-1 就需要单调队列

单调队列优化

- 什么是单调队列优化呢?
- dp[i] = min {f[k]} + x[i]
- 考虑一下两个可选择量j和k，如果j>k而且f[j] < f[k]那么显然可以忽略f[k]的值，就可以舍弃这样的k
- 所以每次只需要考虑j<k 且 f[j] < f[k]的情况（想一想为什么要记录比他大的量？）
- 注意到每个量入队一次出队一次，所以总复杂度为 O(N)

股票交易

- 第i天的股票买入价为每股APi，第i天的股票卖出价为每股BPi，APi>=BPi
- 每天不能无限制地交易，于是股票交易所规定第i天的一次买入至多只能购买ASi股，一次卖出至多只能卖出BSi股。
- 股票交易所规定在两次交易之间，至少要间隔W天。在第i天发生了交易，从第i+1天到第i+W天，均不能发生交易。
- 任何时间，一个人的手里的股票数不能超过MaxP。
- 求T天内最多可以赚多少钱，认为手里的钱是无限多的

股票交易

- dp[i][j]表示第i 天拥有 j只股票的时候，赚了多少钱
- 从前一天不买不卖:
  - dp[i][j]=max(dp[i-1][j],dp[i][j])
- 从前i-W-1天买进一些股:
  - dp[i][j]=max(dp[i-W-1][k]-(j-k)*AP[i],dp[i][j])
- 从i-W-1天卖掉一些股:
  - dp[i][j]=max(dp[i-W-1][k]+(k-j)*BP[i],dp[i][j])

股票交易

- 买入股票：
- dp[i][j]=max(dp[i-W-1][k]+k*AP[i])-j*AP[i]
- 令f[i-W-1][k]=dp[i-W-1][k]+k*AP[i]
- dp[i][j]=max(f[i-W-1][k]) - j*AP[i]
- 卖出股票同理
- 单调队列优化

单调队列优化多重背包

- 多重背包
  - 每个物品价格为Ci，价值为Wi，数量为Li
  - N个物品，M块钱，能获得的最大价值?
- 二进制展开 O(M * sum{log(Li)})
- 单调队列优化 O(NM)

决策单调性优化

- 什么是决策单调性?

- 如果记录每个DP[i]所取值的上一步状态为DP[j]，记P[i] = j，如果P数组是单调的，那么就具有决策单调性

- 决策单调性优化的实现

- 单调栈 —— 决策一定是逐渐变优的

决策单调性优化

- 假设只考虑前i个决策，一定是：
- [1…1][2..2]…[i…i]
- 当加入第i+1个决策的时候，不会出现i+1和i+1之间出现其他元素，一定是从后往前一段连续的区间
- 只需要二分一下断开位置

决策单调性优化

- 有N株植物排成一列，编号1到N，刚开始高度都为0。
- 然第i株植物每天都会生长Li
- 给定一个时间序列长度为M，Ti Bi，表示到第T天将所有超过Bi的植物砍到只有Bi的高度，收集砍掉的部分，求每个时间序列的操作之后，可以收集到的植物高度之和。
- N<=10^5
- M<=10^5   T <= 10^9

斜率优化

- DP[i] = max / min {F[j] + X[i] * Y[j]} + Ci
- 等价于考虑 DP[i] = F[j] + X[i] * Y[j] 其中X[i]为常量（对i而言）
- F[j] = X[i] * Y[j] - DP[i] <==> y = k * x + b
- (x,y)已知，k已知，求最大 / 最小的b

斜率优化

- DP[i] = max / min {F[j] + X[i] * Y[j]} + Ci
- 等价于考虑 DP[i] = F[j] + X[i] * Y[j] 其中X[i]为常量（对i而言）
- F[j] = X[i] * Y[j] - DP[i] <==> y = k * x + b
- (x,y)已知，k已知，求最大 / 最小的b
- 这样的b一定在凸包上

斜率优化

- 用一个栈来维护 （类似Groham求凸包的方法）对应的(x, y)

最小代价

- 给定N个数字，将其划分成若干块后打包（不改变想对顺序），代价为这一块数字之和的平方+M，求划分的最小代价。

最小代价

- f[i] = min(f[j] + cost(j+1,i)) = min (f[j] + (s[i] - s[j])^2 + M)

- f[i] = (f[j] + s[j] * s[j])   - 2 * s[j] * s[i] + (s[i] * s[i] + M)

斜率优化

- 树结构的转移
- 与线性结构不同的是，某一个点如果不是当前凸包上的点，可能是另一分支凸包上的点，不能直接删去

斜率优化

- 树结构的转移
- 与线性结构不同的是，某一个点如果不是当前凸包上的点，可能是另一分支凸包上的点，不能直接删去
- 保留树的性质，建一个指向性边（i —> 凸包上的前一个点，并非树上的父亲）

四边形不等式优化

- 对于区间DP的情况，f[l][r] 的决策 w[l][r] 满足条件：
  - w[l+1][r] <= w[l][r] <= w[l][r-1]
- 按照区间长度DP，对于状态的枚举只要从 w[l+1][r] ~ w[l][r-1]， 记录一下所取的状态为w[l][r]
- 通常打表找出这样的规律
- 很少见