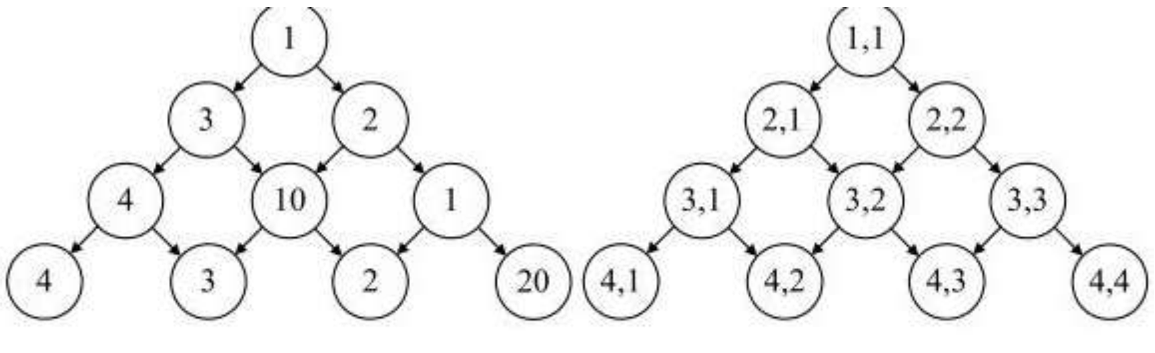
DP(20180129)

问题1： 数字三角形。

如图（a）所示，有一个由非负整数组成的三角形，第一行只有一个数，除了最下行之外，每个数的左下方和右下方各有一个数。从第一行的数开始，每次可以往左下或右下走一格，直到走到最下行，把沿途经过的数全部加起来。如何走，可使得这个和最大?

分析：这是一个多段图上的最短路径问题，其中每行是一个阶段。设d(i,j)为从格子(i,j)出发能得到的最大和，则d(i,j) = a(i,j) + max{d(i+1,j)，d(i+1,j+1)} ，边界是d(n+1,j) = 0，各个格子的编号如图（b）所示。



（a） （b）

问题 2：　嵌套矩形

有n个矩形，每个矩形可以用两个整数a,b描述，表示它的长和宽。矩形X(a,b)可以嵌套在矩形Y(c,d)中的条件为：当且仅当a<c,b<d,或者b<c,a<d(相当于把矩形X旋转90°)。例如,矩形(1,5)可以嵌套在矩形(6,2)内,但不能嵌套在矩形(3,4)内。选出尽量多的矩形排成一行,使得除了最后一个之外,每一个矩形都可以嵌套在下一个矩形内。

分析：本题是DAG最长路问题。设d(i)为以矩形i结尾的最长链的长度,则$d(i) = \max\{0, d(j) \mid 矩形j可以嵌套在矩形i中\}+1$。

问题3：硬币问题

有n种硬币，面值分别为V1,V2,…,Vn,每种都有无限多。给定非负整数S,可以选用多少个硬币,使得面值之和恰好为S？输出硬币数目的最小值和最大值。其中,$1\leqslant n\leqslant 100$,$0\leqslant S\leqslant 10000$,$1\leqslant Vi\leqslant S$。

分析：本题是DAG最长路和最短路问题。设f(i)和g（i）分别为面值之和恰好为i时，硬币数目的最小值和最大值，则

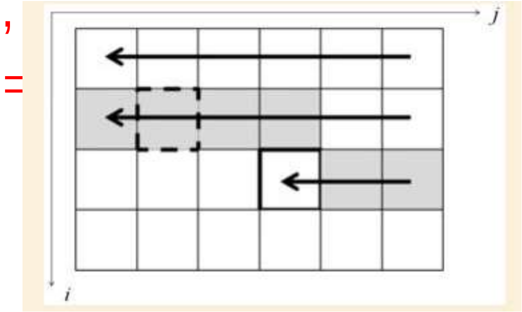
1. $f(i) = \min(\infty, f(i-V_j+1)+1 \mid V_j\leqslant i\}$
2. $g(i)=\max(-\infty, g(i-V_j+1)+1 \mid V_j\leqslant i\}$
3. 边界条件是$f(0)=0$, $g(0)=0$

问题4： 01背包问题

有n种物品，每种只有一个。第i种物品的体积为Vi，重量为Wi。选一些物品装到一个容量为C的背包，使得背包内物品在总体积不超过C的前提下重量尽量大。其中，$1 \leqslant n \leqslant 100$，$1 \leqslant Vi \leqslant C \leqslant 10000$，$1 \leqslant Wi \leqslant 10^6$。

【分析】

1. f(i,j)表示“把前i个物品装到容量为j的背包中的最大总重量”，
2. $f(i,j) = \max\{f(i-1,j),\ f(i-1,j-V_i) + Wi \mid Vi \leqslant j\}$，边界为f(0,j) =
3. 可以使用滚动数组优化空间。



```
for(int i = 1; i <=n; i++)
  for(int j = C; j >=0; j--)
    if(j >= V_i) f[j] = max(f[j], f[j-V_i]+W_i);
```

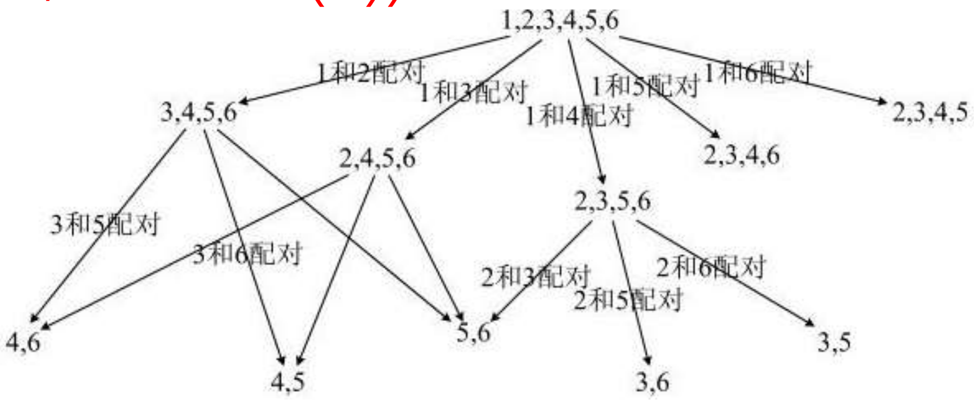
考虑一下其中的依赖关系。

问题5： 点集配对问题

空间里有n个点P_0,P_1,P_{n-1},把它们配成n/2对(n是偶数)，使得每个点恰好在一个点对中。要求所有点对中,两点的距离之和应尽量小。其中n≤20,|xi|,|yi|,|zi|≤10000。

【分析】

1. $d(S) = \min(|Pi-Pj| + d(S – \{i,j\}) \mid j \text{ in } S, i = \min(S))$
2. $d(\varnothing) = 0$
3. 决策考虑S中最小的i和谁配对呢?
4. $O(2^n n)$
5. 位集合运算



问题6: 最长上升子序列问题（LIS）

给定n个整数$A_{1\cdots n}$，按从左到右的顺序选出尽量多的整数，组成一个上升子序列（子序列可以理解为：删除0个或多个数，其他数的顺序不变）。比如，从序列1,6,2,3,7,5中，可以选出上升子序列1,2,3,5。也可以选出1,6,7但前者更长。选出的上升子序列中相邻元素不能相等。

【分析】

1. 以i为结尾的LIS长度d(i)，则$d(i)=\max\{d(j)+1 \mid j < i, A_j<A_i\}$, 复杂度$O(n^2)$，如果n太大($10^5$)，完全扛不住。
2. 考虑优化:
3. 两个长度都为k的LIS，结尾元素更小的那个明显有可能变得更长。
4. 定义$g(k)=\min\{A_j \mid d[j] = k\}$，初始$g(k) = \infty$
5. $g(1)\leqslant g(2) \leqslant\cdots \leqslant g(n)$，g值单调递增。
6. 动态维护g数组

```
for(int i = 1; i <= n; i++) g[i] = INF;
int ans = 0;
for(int i = 0; i < n; i++) {
    int k = lower_bound(g+1, g+n+1, S[i]) - g;
    d[i] = k;
    g[k] = S[i];
    ans = max(ans, d[i]);
}
```

问题7: 最长公共子序列问题（LCS）

给出两个子序列A和B,求长度最大的公共子序列

分析

1. 设d(i,j)为A[1…i]和B[1…j]的LCS长度,
2. 则当A[i] = B[j]时, d(i,j)=d(i-1,j-1)+1
3. 否则d(i,j) = max{d(i-1,j),d(i,j-1)}
4. O(nm)
5. 其中n和m分别是序列A和B的长度。LCS 问题也可以用**滚动数组法**进行优化
6. 二维滚动数组，　D[2,N]

问题8: 最大连续和

给出一个长度为n的序列$A_{1\cdots n}$，求一个连续子序列$A_{i\cdots j}$，使得元素总和$A_{i\cdots j}$最大。

1. 前缀和
2. 分治法
3. DP,d(i)表示以i为结尾的最大连续和,$d(i)=max(0,d(i-1))+A_i$

问题9: 货郎担问题（TSP）

有n个城市，两两之间均有道路直接相连。给出每两个城市i和j之间的道路长度$L_{i,j}$,求一条经过每个城市一次且仅一次,最后回到起点的路线, 使得经过的道路总长度最短。其中n≤15,城市编号为0~n-1。

【分析】

当前所在的位置i，还未走的城市集合S，暴力枚举下一步的城市j

1. $d(i, S) = \min(L_{ij}+d(j, S+\{j\}) \mid j \notin S)$
2. $d(i, \varnothing) = L_{i,0}$
3. 目标值$d(0, S-\{0\})$
4. $O(n*2^n)$

问题10：矩阵链乘(MCM)

一个n×m矩阵由n行m列共n×m个数排列而成。两个矩阵A和B可以相乘的条件为：当且仅当A的列数等于B的行数。一个n×m的矩阵乘以一个m×p的矩阵等于一个n×p的矩阵，运算量为mnp。

矩阵乘法 不满足分配律，但满足结合律，因此A×B×C既可以按顺(A×B)×C进行，也可以按A×(B×C)来进行。假设A、B、C分别是2×3，3×4和4×5矩阵，则(A×B)×C的运算量为2×3×4＋2×4×5＝64，A×(B×C)的运算量为3×4×5＋2×3×5＝90。显然, 第一种运算顺序更节省运算量。给出n个矩阵组成的序列，设计一种方法把它们依次相乘，使得总运算量最小。假设第i个矩阵Ai是pi－1×pi的。

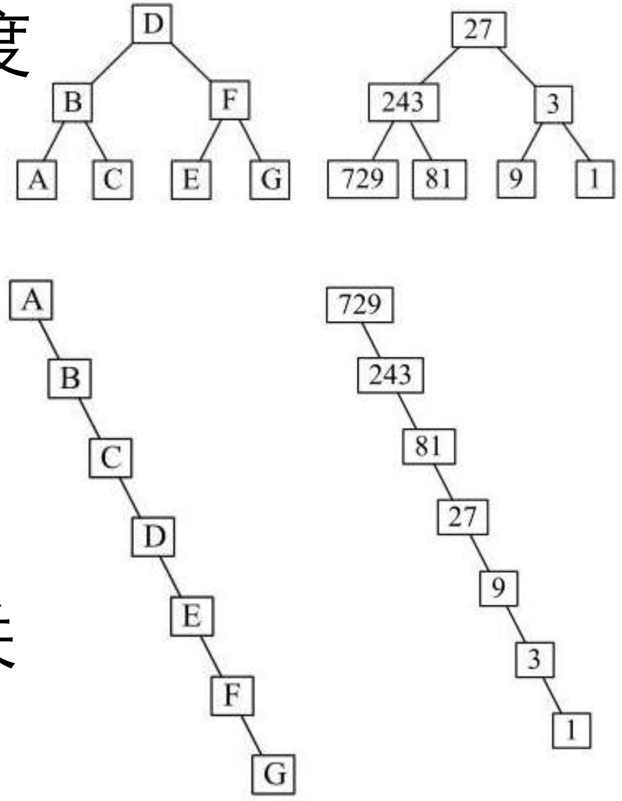
【分析】

1. 设 f(i,j)表示把$A_{i\dots j}$乘起来所需要的乘法次数
2. 枚举“最后一次乘法”是第k个乘号
3. $f(i,j) = \min\{f(i,k) + f(k+1,j)+p_{i-1}p_kp_j\}$
4. 边界f(i,i)=0，时间复杂度为$O(n^3)$

问题11:最优排序二叉树问题（OBST)

给n个符号建立一棵排序二叉树[6]。虽然平衡树的高度最小，但如果各个符号的频率相差很大，平衡反而不好。比如，若有7个符号ABCDEFG，频率分别为729,243,81,27,9,3,1，平衡树的总检索次数（即所有关键字频率和深度的乘积之和）为 $27\times1+(243+2)\times2+(729+81+9+1)\times3 = 2977$。

链状树总检索次数仅为1636 次。给定n个关键字的频率$f_{1\cdots n}$，要求构造一棵最优的排序二叉树，使得每个关键字的频率和深度的乘积之和最小。



OBST

每次选树根，递归建立左右子树。

1. 状态点集[i…j]->总检索次数d(i,j)
2. 枚举树根k, $d(i,j)=f_k+d(i,k-1)+d(k+1,j)+\Sigma f_{i\cdots k-1}+\Sigma f_{k+1\cdots j}=d(i,k-1)+d(k+1,j)+\Sigma f_{i\cdots j}$
3. 边界$d(i,i)=f_i$
4. 状态n^2个，决策n次，复杂度$O(n^3)$
5. 四边形不等式优化?

树的重心

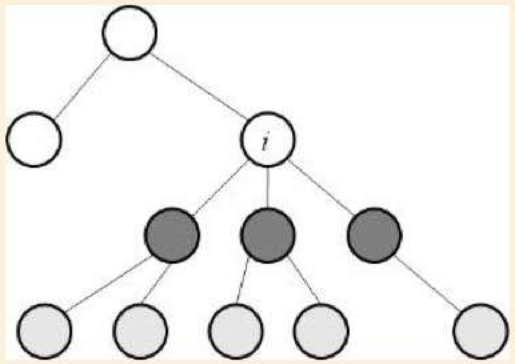


图9-12 结点i的gs(i)（浅灰色）和s(i)（深灰色）

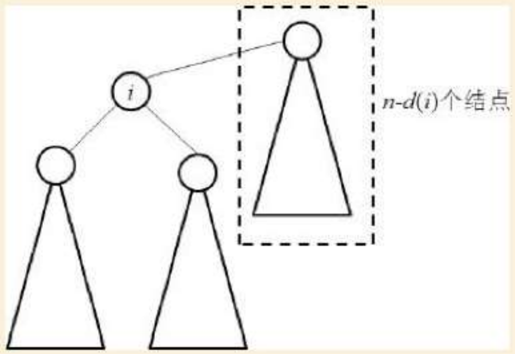


图9-13 树中的结点分布

1. d(u)为u子树的结点数
2. d(u) = Σd(v) + 1
3. f(u) = max(d(v), u父子树个数->n-d(u))
4. u = argmin(f(u))

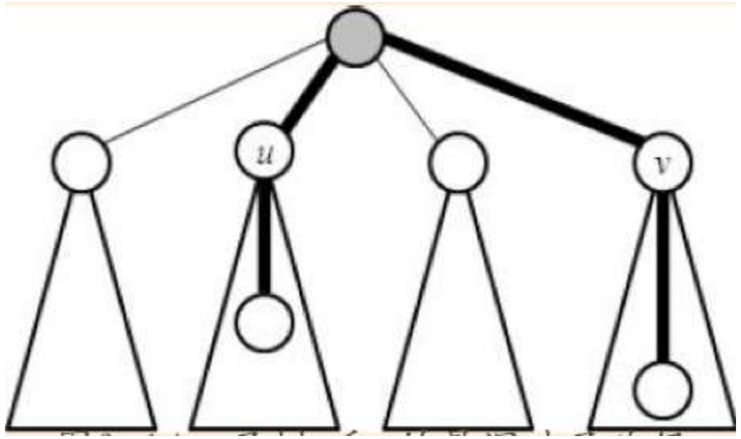
树的最大独立集

正着求和不容易，不如反着来刷表。

$$d(i)=\max\{1+\sum_{j\in gs(i)} d(j), \sum_{j\in s(i)} d(j)\}$$

树的最长路径（最远点对）

对于一棵 n 个结点的无根树，找到一条最长路径。换句话说，要找到两个点，使得它们的距离最远。



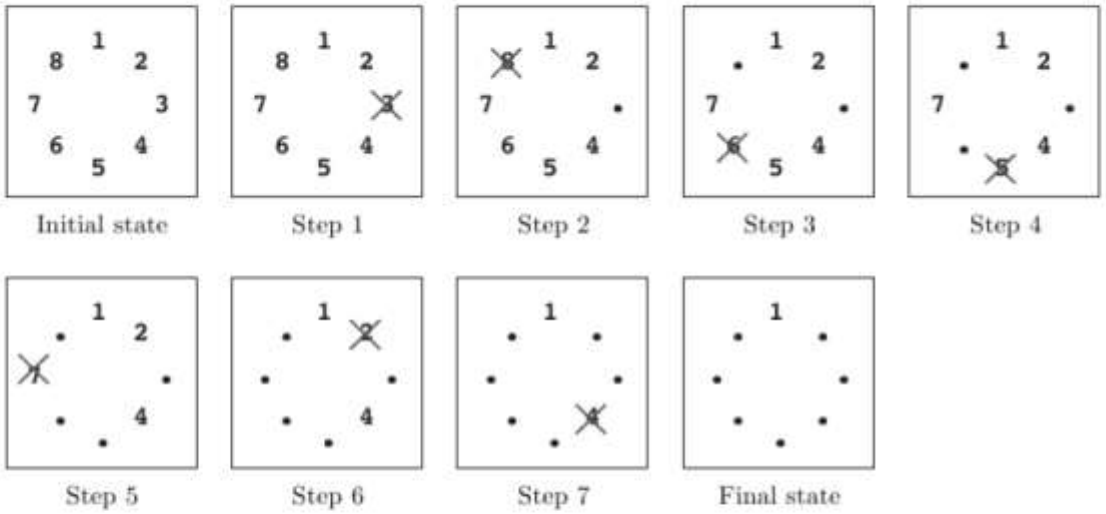
图的色数

图论 有一个 经典 问题是 这样 的： 给 一个 无向 图 G， 把 图中的 结点 染成 尽量 少的 颜色， 使得 相邻 结点 颜色 不同。

1. 染色影响一片? 怎么办?
2. 影响哪一片? 不影响哪一片?
3. 每次染互相不连通的一个子集。颜色数+1，集合减去一个子集

例题5-26 约瑟夫问题的变形(And Then There Was One, Japan 2007, LA 3882)

n 个数排成一个圈。第一次删除m，以后每数k个数删除一次，求最后一个被删除的数。当n=8,k=5,m=3时，删数过程如图所示。



【分析】

1. 定义问题f(n): 数字n个数0,1,…n-1，形成一个环，从0开始每k个数删一个，最后剩下几?
2. 考虑0,1,2,k-1,k..n, 删除k-1之后，将k(可能>n)编码为0, 则f(n) = (f(n-1)+k)%n, f(1) = 0
3. 一开始将数字m-k编码为0，则最终答案为(m-k+f(n)) % n，注意k > n的情况处理
4. 最后再将结果+1

例题5-27 王子和公主(Prince and Princess, UVa 10635)

有两个长度分别为$p+1$和$q+1$的序列，每个序列中的各个元素互不相同，且都是$1\sim n^2 (n \leqslant 250)$之间的整数。两个序列的第一个元素均为1。求出A和B的LCS长度。

【分析】

1. 典型的LCS问题, 但是p,q很大(n^2)，最终复杂度为$O(n^4)$
2. 如何做优化?
3. 注意到AorB中的序列元素都不同，考虑将B中每个元素转换成其在A中的位置(没有，则为0)
4. 则A&B的LCS就变成求B的LIS，加上之前说的优化

例题5-28 Sum游戏(Game of Sum, UVa10891)

有一个长度为n的整数序列， 两个游戏者A和B轮流取数，A先取。每次玩家只能从左端或者右端取一个数，但不能两端都取。所有数都被取走后游戏结束，然后统计每个人取走的所有数之和，作为各自的得分。两个人采取的策略都是让自己的得分尽量高，并且两人都足够聪明，求A的得分减去B的得分后的结果。

【分析】

1. 博弈游戏类的状态，考虑先手面对局面采取最有策略的最大得分。
2. 局面就是A的子区间A[i,j]，定义D(i,j)=足够聪明的先手玩家面对这个子区间时的最大得分。$S[i]=\Sigma A_{1\cdots i}$
3. 足够聪明的玩家会采取最优策略使得：下一个玩家面对剩余局面时的最大得分最小化
4. 状态转移方程参见代码, $O(n^3)$
5. 优化: f(i,j)=min(dp(i,k)), k in [i,j-1], g(i,j) = min(dp(k, j)), k in [i+1,j]

```
s = S[j] - S[i-1];
d = s;
_for(k, i, j){ // [i,k], [k+1,j]
  d = max(d, s - dp(k+1,j));
  d = max(d, s - dp(i,k));
}
```

！！！ 例题5-29 黑客的攻击(Hacker's Crackdown, UVa11825)

假设你是一个黑客，侵入了一个有着n≤16台计算机（编号为0,1,⋯,n－1）的网络。一共有n种服务，每台计算机都运行着所有服务。对于每台计算机，你都可以选择一项服务，终止这台计算机和所有与它相邻计算机的该项服务(如果其中一些服务已经停止，则这些服务继续处于停止状态)。你的目标是让尽量多的服务器完全瘫痪(即：没有任何计算机运行该项服务)。

【分析】

1. 数学模型，集合A{0⋯n-1}有很多子集。$P_0,P_1 \cdots P_{n-1}$，七中Pi就是点i及其邻居组成的集合
2. 需要给每个Pi确定一个分组G(停哪个服务?)，使得每个G中的Pi之并集就是全集
3. 目标就是分组(完全瘫痪的服务)的个数尽量多
4. n很小，所以可以用位集合表示每个P
5. 定义S为A的子集{s1,s2⋯sk}，则定义Cover(S) = $\cup\{P_{s1},P_{s2}\cdots P_{sk}\}$

例题9-1 城市里的间谍(A Spy in the Metro, ACM/ ICPC World Finals 2003, UVa1025)

某城市的地铁是线性的，有n（2≤n≤50）个车站，从左到右编号为1～n。有M1辆列车从第1站开始往右开，还有M2辆列车从第n站开始往左开。在时刻0，Mario从第1站出发，目的是在时刻T（0≤T≤200）会见车站n的一个间谍。在车站等车时容易被抓，所以她决定尽量躲在开动的火车上，让在车站等待的总时间尽量短。列车靠站停车时间忽略不计，且Mario身手敏捷，即使两辆方向不同的列车在同一时间靠站，Mario也能完成换乘。

【分析】

1. 时间序
2. 状态: {t, 车站} -> 最长等待多久
3. 决策:
    1. 等1分钟
    2. 往左开
    3. 往右开
4. 边界状态
5. 时间复杂度D(n*T) -> 怎么计算的?

例题9-2 巴比伦塔(The Tower of Babylon, UVa437)

有 n(n≤30)种立方体,每种都有无穷多个。要求选一些立方体摞成一根尽量高的柱子（可以自行选择哪一条边作为高），使得每个立方体的底面长宽分别严格小于它下方立方体的底面长宽。

【分析】

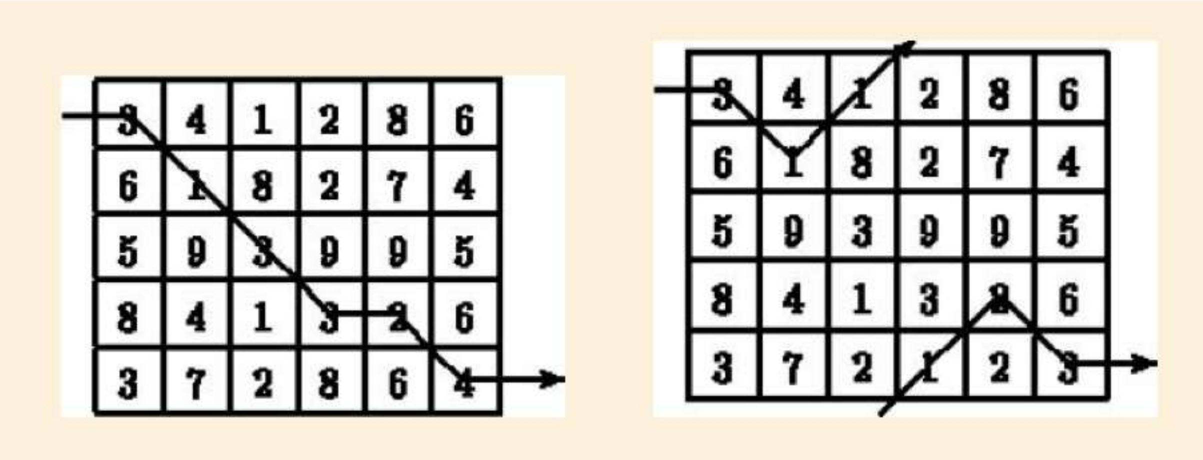
1. 状态(顶部立方体编号i, 高编号hi(0->3))
2. DAG
3. 解(立方体上面叠加一个立方体)
4. 总结 -> 状态简化压缩

例题9-4 单向TSP(UnidirectionalTSP,UVa116)

给一个m行n列(m≤10,n≤100)的整数矩阵，从第一列任何一个位置出发每次往右、右上或右下走一格，最终到达最后一列。要求经过的整数之和最小。整个矩阵是环形的，即第一行的上一行是最后一行，最后一行的下一行是第一行。输出路径上每列的行号。多解时输出字典序最小的。图中是两个矩阵和对应的最优路线(唯一的区别是最后一行)。

【分析】

1. 序: 当前列，从小到大
2. 状态:（列，行)
3. 初始:dp(0, (1->R))
4. 记忆化 & 递推
5. 字典序问题



例题9-5 劲歌金曲(Jin Ge Jin Qu hao, UVa12563)

如果问一个麦霸："你在KTV里必唱的曲目有哪些？"得到的答案通常都会包含一首"神曲"：古巨基的《劲歌金曲》。为什么呢？一般来说，KTV不会在"时间到"的时候鲁莽地把正在唱的歌切掉，而是会等它放完。例如，在还有15秒时再唱一首2分钟的歌，则实际上多唱了105秒。但是融合了37首歌曲的《劲歌金曲》长达11分18秒(5)，如果唱这首，相当于多唱了663秒！假定你正在唱KTV，还剩t秒时间。你决定接下来只唱你最爱的n首歌（不含《劲歌金曲》）中的一些，在时间结束之前再唱一个《劲歌金曲》，使得唱的总曲目尽量多（包含《劲歌金曲》），在此前提下尽量晚的离开KTV。输入n（n≤50），t（$t \leq 10^9$）和每首歌的长度（保证不超过3分钟(6)），输出唱的总曲目以及时间总长度。输入保证所有n＋1首曲子的总长度严格大于t。

【分析】

1. 选择J是永远是更优的。所以要留一些时间给J。
2. 歌曲长度如何排序?
3. 时间不超过T的前提下:D(i,t)->当前决策到i, 已经用了t分钟，最多能唱多少。F(i,t)最多能唱多久。
4. 每首歌唱与不唱，分别进行决策, 01背包。

例题9-6 照明系统设计(Lighting System Design, UVa11400)

你的任务是设计一个照明系统。一共有n（n≤1000）种灯泡可供选择，不同种类的灯泡必须用不同的电源，但同一种灯泡可以共用一个电源。每种灯泡用4个数值表示：电压值V（V≤132000），电源费用K（K≤1000），每个灯泡的费用C（C≤10）和所需灯泡的数量L（1≤L≤100）。假定通过所有灯泡的电流都相同，因此电压高的灯泡功率也更大。为了省钱，可以把一些灯泡换成电压更高的另一种灯泡以节省电源的钱（但不能换成电压更低的灯泡）。你的任务是计算出最优方案的费用。

【分析】

1. 排序，电压从小到大。
2. 为什么只换一部分不划算?Ci->Cj，Ci更贵，电源钱没省，不如全部换了，省电源还省灯泡。如果Cj更贵，如果只换一部分，电源没省，灯泡更贵。
3. d[i]= min{ d[j] ＋ (s[i] – s[j])* c[i] ＋ k[i])} [j,i]区间全部升级到i。

例题9-7 划分成回文串(Partitioning by Palindromes, UVa 11584)

输入一个由小写字母组成的字符串，你的任务是把它划分成尽量少的回文串。例如，racecar本身就是回文串；fastcar只能分成7个单字母的回文串，aaadbccb最少分成3个回文串：aaa,d,bccb。字符串长度不超过1000。

【分析】

1. 暴力枚举最后一刀
2. 切分DP
3. 预处理

例题9-9切木棍(Cutting Sticks, UVa 10003)

有一根长度为L(L<1000)的棍子，还有n(n < 50)个切割点的位置（按照从小到大排列)。你的任务是在这些切割点的位置处把棍子切成n＋1部分, 使得总切割费用最小。每次切割的费用等于被切割的木棍长度。例如, L=10, 切割点为2,4,7。如果按照2,4,7的顺序, 费用为10+8+6=24，如果按照4,2,7的顺序, 费用为10＋4＋6=20。

【分析】

1. 暴力枚举第一刀
2. 切分DP
3. d= min(C[r]-C[l]+dp(l,i)+dp(i,r), d);
4. 预处理

例题9-10 括号序列(Brackets Sequence, NEERC2001,UVa1626)

定义如下正规括号序列(字符串,简称正规)：空序列是正规。如果S是正规，那么(S)和[S]也是正规。如果A和B都是正规，那么AB也是正规。例如,下面的字符串都是正规：()，[]，(())，([])，()[]，()[()]，而如下字符串则不是正规：(，[，]，)(，([()。输入一个长度不超过100的，由“(”、“)”、“[”、“]”构成的序列，添加尽量少的括号，得到一个正规序列。如有多解，输出任意一个序列即可。

【分析】

如果 S 形如(S′) 或者[S′], 转移到d(S′)。
如果 S至少有两 个 字符， 则 可以 分成 AB， 转移 到 d(A) + d(B)。

例题9-12 工人的请愿书(Another Crisis, UVa12186)

某公司里有一个老板和n（$n \leq 10^5$）个员工组成树状结构，除了老板之外每个员工都有唯一的直属上司。老板的编号为0，员工编号为1～n。工人们（即没有直接下属的员工）打算签署一项请愿书递给老板，但是不能跨级递，只能递给直属上司。当一个中级员工（不是工人的员工）的直属下属中不小于T%的人签字时，他也会签字并且递给他的直属上司。

【分析】

1. 设d(u)表示让u给上级发信最少需要多少个工人
2. 假设u有k个子结点，则至少需要c=(kT-1)/100+1个直接下属发信才行
3. 把所有子结点的d值从小到大排序，前c个加起来即可
4. 简单贪心背包

例题9-13 Hali-Bula的晚会(Party at Hali-Bula,ACM/ICPCTehran2006,UVa1220)

公司里有n（n≤200）个人形成一个树状结构，即除了老板之外每个员工都有唯一的直属上司。要求选尽量多的人，但不能同时选择一个人和他的直属上司。问：最多能选多少人，以及在人数最多的前提下方案是否唯一。

【分析】

1. d(u,0)和f(u,0)表示以u为根的子树中，不选u点能得到的最大人数以及方案唯一性f=1表示唯一，0表示不唯一）。
2. d(u,1)和f(u,1)表示以u为根的子树中，选u点能得到的最大人数以及方案唯一性。
3. d(u,1)的计算：
    1. u的子结点v都不能选，d(u,1)=Σ{d(v,0)|v是u的子结点}。当且仅当所有f(v,0)=1时f(u,1)才是1。
4. d(u,0)的计算：
    1. v可选可不选，即d(u,0)=Σ{max(d(v,0),d(v,1))}。
    2. 如果某个d(v,0)和d(v,1)相等，则不唯一
    3. 如果max取到的那个d(v,*)对应的f=0，方案也不唯一(如d(v,0)>d(v,1)且f(v,0)=0，则f(u,0)=0)。

例题9-14 完美 的 服务(Perfect Service, ACM/ ICPC Kaoshiung 2006, UVa1218)

有n（n≤10000）台机器形成树状结构。要求在其中一些机器上安装服务器，使得每台不是服务器的计算机恰好和一台服务器计算机相邻。求服务器的最少数量。如图9-15所示，图a是非法的，因为4同时和两台服务器相邻，而6不与任何一台服务器相邻。而图b是合法的。

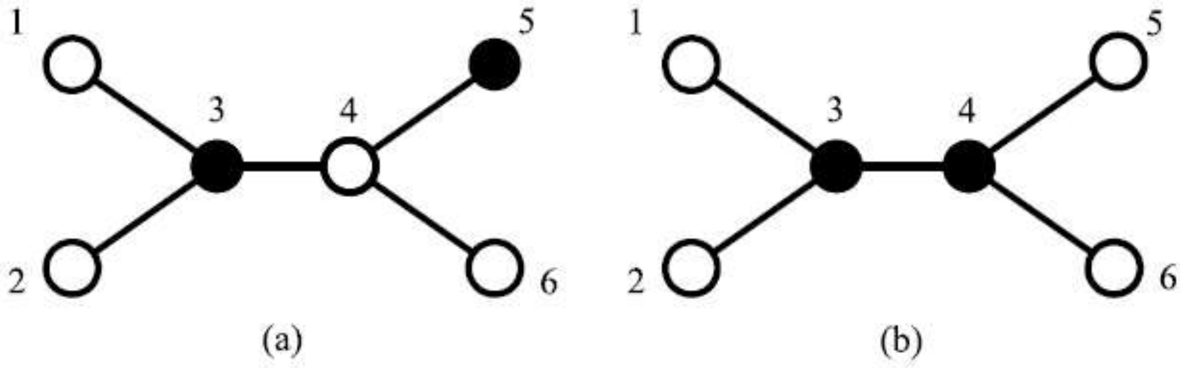


Figure 1

【分析】

1. 有根树，每个子树树根进行决策
2. d(u,0)：u是服务器，则每个子结点v可以是服务器也可以不是。
3. d(u,1)：u不是服务器，但u的父亲是服务器，这意味着所有v都不是服务器。
4. d(u,2)：u和u的父亲都不是服务器。这意味着恰好有一个v是服务器。

例题9-15 校长的烦恼(Headmaster's Headache,UVa10817)

某校有m个教师和n个求职者，需讲授s个课程(1≤s≤8, 1≤m≤20, 1≤n≤100)。已知每人的工资c(10000≤c≤50000)和能教的课程集合，要求支付最少的工资使得每门课都至少有两名教师能教。在职教师不能辞退。

【分析】

1. 状态: 课程状态(位集合)，n个求职者依次决策
2. m和n个求职者统一处理

习题9-2 免费糖果(Free Candies, UVa10118)

桌上有4堆糖果，每堆有N（N≤40）颗。佳佳有一个最多可以装5颗糖的小篮子。他每次选择一堆糖果，把最顶上的一颗拿到篮子里。如果篮子里有两颗颜色相同的糖果，佳佳就把它们从篮子里拿出来放到自己的口袋里。如果篮子满了而里面又没有相同颜色的糖果，游戏结束，口袋里的糖果就归他了。当然，如果佳佳足够聪明，他有可能把堆里的所有糖果都拿走。为了拿到尽量多的糖果，佳佳该怎么做呢？

状态关联的时候要考虑压缩合并。

习题 9- 3　　切 蛋糕 （ Cake Slicing, ACM/ ICPC Nanjing 2007, UVa1629）

有一个n行m列（1≤n，m≤20）的网格蛋糕上有一些樱桃。每次可以用一刀沿着网格线把蛋糕切成两块，并且只能够直切不能拐弯。要求最后每一块蛋糕上恰好有一个樱桃，且切割线总长度最小。如图9-30所示是一种切割方法。

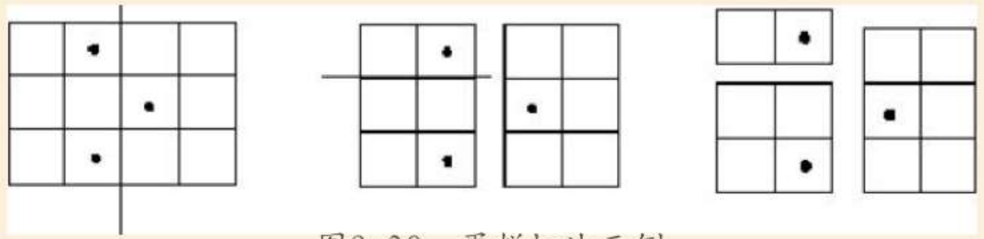


图9-30　蛋糕切法示例

1. 暴力想到DP
2. DP套DP

习题9-4 串 折叠 （Folding, UVa1630）

给出一个由大写字母组成的长度为n（1≤n≤100）的串，“折叠”成一个尽量短的串。例如，AAAAAAAAAABABABCCD折叠成9(A)3(AB)CCD。折叠是可以嵌套的，例如，NEERCYESYESYESNEERCYESYESYES可以折叠成2(NEERC3(YES))。多解时可以输出任意解。

1. 区间DP
2. 暴力搜索各种分治方案
3. 回溯->记忆化

习题9-6 电子人的基因(Cyborg Genes, UVa 10723)

输入两个A~Z组成的字符串（长度均不超过30），找一个最短的串，使得输入的两个串均是它的子序列（不一定连续出现）。你的程序还应统计长度最短的串的个数。例如，ABAAXGF和AABXFGA的最优解之一为AABAAXGFGA，一共有9个解。

1. 拼字符串
2. 定义D[i,j]为S1[i…n1]和S2[j…n2]的最优解
3. 每一步从左边选择，S1[i]=S2[j] 则没问题，不等则考虑两种决策方案。
4. 回溯->记忆化

习题 9-7 密码锁（Locker, Tianjin 2012, UVa1631）

有一个n（n≤1000）位密码锁，每位都是0～9，可以循环旋转。每次可以让1～3个相邻数字同时往上或者往下转一格。例如，567890->567901（最后3位向上转）。输入初始状态和终止状态（长度不超过1000），问最少要转几次。例如，111111到222222至少转2次，由896521到183995则要转12次。

1. 从左到右决策
2. 因为可能一次操作影响三位数字， 所以状态就考虑这3个数字嘛
3. 回溯->记忆化搜索

习题 9-8 阿里巴巴（Alibaba, ACM/ ICPC SEERC 2004, UVa1632）

直线上有n（n≤10000）个点，其中第i个点的坐标是xi，且它会在di秒之后消失。Alibaba可以从任意位置出发，求访问完所有点的最短时间。无解输出No solution。

【分析】

1. 所有已经访问过的点形成一个连续区间，否则不合算
2. 状态? 区间的左右端点，以及当前在左右。
3. 决策，往左走，往右走
4. 滚动数组递推(n^2 = 10e8)
5. 递推顺序，从小到大

习题9-9 仓库守卫(Storage Keepers, UVa10163)

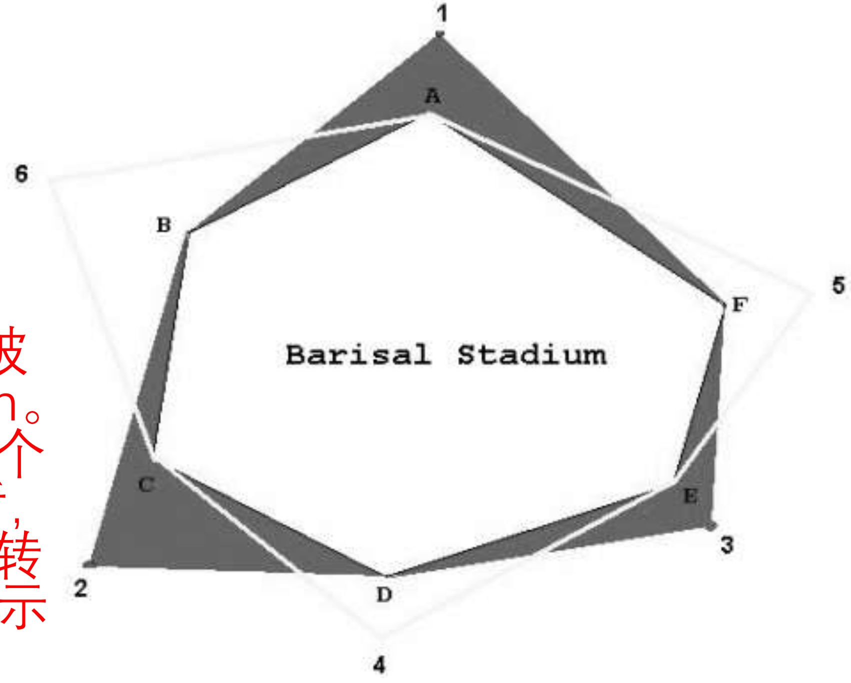
你有n(n≤100)个相同的仓库。有m(m≤30)个人应聘守卫，第i个应聘者的能力值为P_i(1≤P_i≤1000)。每个仓库只能有一个守卫，但一个守卫可以看守多个仓库。如果应聘者i看守k个仓库，则每个仓库的安全系数为P_i/K的整数部分。没人看守的仓库安全系数为0。

你的任务是招聘一些守卫，使得所有仓库的最小安全系数最大，在此前提下守卫的能力值总和（这个值等于你所需支付的工资总和）应最小。

1. DP1: 决策顺序，每个守卫依次决策
2. 状态 {仓库个数，守卫编号} -> 最小安全系数
3. DP2: 决策顺序相同
4. 状态 {仓库个数，守卫编号} -> 满足DP1的费用

习题9-10 照亮体育馆(Barisal Stadium, UVa10641)

输入一个凸n(3<=n<=30)边形体育馆和多边形外的m(1<=m<=1000)个点光源，每个点光源都有一个费用值。选择一组点光源，照亮整个多边形，使得费用值总和尽量小。如下图，多边形ABCDEF可以被两组光源{1,2,3}和{4,5,6}照亮。光源的费用决定了哪组解更优。



1. 环形区间(1~n) ->(1-2n)
2. 预处理每个光源可以照亮的点
3. 考虑每个[i,i+n]的区间
4. 对于顶点 i(0≤i<n)：定义 D(j)为顶点编号区间[i,j)内的顶点都被照射到所需的最小费用。 则本题的解就是 min(D(i+n)), 0≤i<n。对于每个 i，从小到大遍历顶点编号 j(i≤j<i+n)，然后考 虑每个能照射到 j 的光源 lt，记 r = min(lt.R,i+n)，表示使用了 lt 之后，能够照射到的右边界。分 是否使用 lt 两种情况考虑进行状态转移，用 D(j)更新 D(r)：D(r) = min(D(r), D(j) + lt.c)，其中 lt.c 表示光源 lt 的费用。
5. 典型刷表法

习题9-11 禁止的回文子串(Dyslexic Gollum, ACM/ICPC Amritapuri 2012, UVa1633)

输入正整数n和k(1<=n<=400, 1<=k<=10)，求长度为n的01串中有多少个不含长度至少为k的回文连续子串。比如n=k=3时只有4个串满足条件：001, 011, 100, 110。

1. 决策顺序: 从左到右
2. 状态(当前位，后k位压缩)
3. 预处理出长度k/k+1的回文，转移时别出现

习题9-13叠盘子(Stacking Plates, ACM/ICPC World Finals 2012, UVa1289)

有n(1<=n<=50)堆盘子，第i堆盘子有h_i个盘子(1<=h_i<=50)，从上到下直径不减。所有盘子的直径均不超过10000。有两种操作:

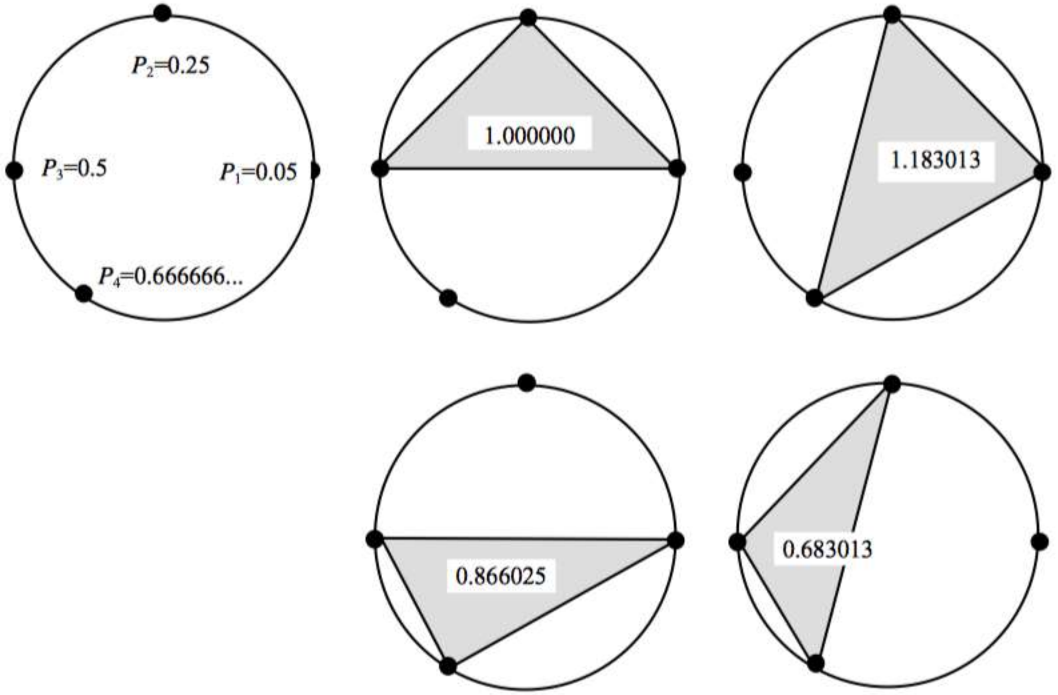
- split：把一堆盘子从某个位置处分成上下两堆。
- join：把一堆盘子a放到另一堆盘子b的顶端，要求是a底部盘子的直径不超过b顶端盘子的直径。
- 你的任务是用最少的操作把所有盘子叠成一堆。

1. 如果操作1进行了x次，则分成了n+x堆，需要n+x-1次合并，所有的操作次数就是2x+n-1
2. 对盘子按照初始所在的堆进行染色，则最终按照大小排序好后，颜色切换的次数是C，那么X = C-n+1=C-(n-1)
3. 最终答案就是2C-N+1，问题转化为求C的最小值
4. 直径离散化处理，同色相同直径的盘子去重。盘子的最大直径是10000，个数最大2500，不同的盘子<=2500个。
5. 记F(d, i)为直径≤d的盘子形成一堆，且底部颜色为i时，最小的C。状态转移就是每次决策下一次往上堆的盘子是哪种颜色。
6. 决策顺序，d从小到大，依次尝试不同的颜色

习题9-14 圆和多边形(Telescope, ACM/ICPC Tsukuba 2000, UVa1543)

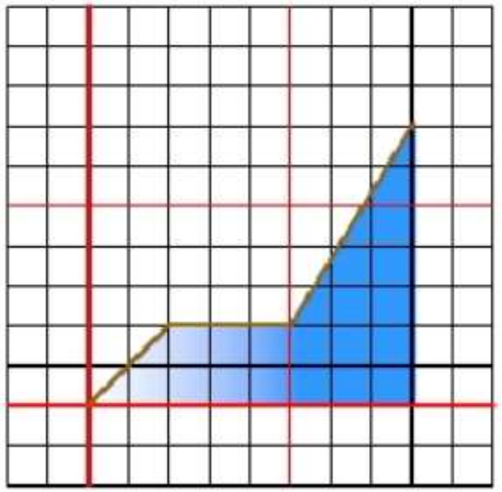
给你一个圆和圆周上的n(3<=n<=40)个不同点。请选择其中的m(3<=m<=n)个，按照在圆周上的顺序连成一个m边形，使得它的面积最大。例如在下图的例子中，右上方的多边形最大。

1. 对点编号
2. 1~n区间，考虑1~n中选择的编号最大的点



习题9-15 学习向量(Learning Vector, ACM/ICPC Dhaka 2012, UVa12589)

输入n个向量(x,y)(0<=x,y<=50)，要求选出k个，从(0,0)开始画，使得画出来的折线与x轴围成的图形面积最大。比如四个向量是(3,5), (0,2), (2,2), (3,0)，可以依次画(2,2), (3,0), (3,5)，围成的面积是21.5。输出最大面积的两倍。1<=k<=n<=50。



1. 按照斜率大到小排序
2. 每个斜线选择与否

习题9-17 佳佳的筷子(Chopsticks, UVa10271)

中国人吃饭喜欢用筷子。佳佳与常人不同，他的一套筷子有三只，两根短筷子加上一只比较长的（一般用来穿香肠之类的食物）。两只较短的筷子的长度应该尽可能接近，但是最长那只的长度是无所谓的。如果一套筷子的长度分别是*A*，*B*，*C*($A \leqslant B \leqslant C$)，则用$(A-B)^2$的值表示这套筷子的质量，这个值越小，这套筷子的质量越高。

佳佳请朋友吃饭，并准备为每人准备一套这种特殊的筷子。佳佳有*N*(*N*<=1000)只筷子，他希望找到一种办法搭配好*K*+8套筷子，使得这些筷子的质量值和最小。保证筷子足够，即3*K*+24<=*N*。

1. 排序
2. 按照编号从小到大决策
3. 任意AB必然是相邻的否则可以交换变成更小的
4. [i,j], 前i个筷子中选出j组的最小值
5. 决策两种: 是否使用(i,i-1)作为AB
6. 从大到小排序
7. i,j <- (i-2,j-1) & (i-1, j)

习题9-18 棒球投手(Pitcher Rotation, ACM/ICPC Kaosiung 2006, UVa1379)

你经营着一支棒球队。在接下来的g+10天中会有g(3<=g<=200)场比赛，其中每天最多一场比赛。你已经分析出你的n(5<=n<=100)个投手中每个人对阵所有m(3<=m<=30)个对手的胜率（一个n*m矩阵），要求给出作战计划（即每天使用哪个投手），使得总获胜场数的期望值最大。注意，一个投手在上场一次后至少要休息4天。

提示：如果直接记录前4天中每天上场的投手编号(1~n)，时间和空间都无法承受。

1. 投手上场一次之后休息4天，前面4天用过的投手今天不能再用
2. 编号n(≤100)，如果记录编号空间复杂度为n^4
3. 记录对阵当天对手前5名(至少有1个可用)的选手的名次
4. 决策: 当天是选第几名

5. 全期望公式
6. 滚动数组