DP 选讲

ppt 太难添加好看的数学公式，也还不会用 $beamer$ ，所以就简单的做一份竖版的讲义。

有任何问题什么时候都可以找我问。

内容会以例题的方式呈现，题不会太难。大纲大概是这样的：

$Pt.1$ 关于 DP

1.1 什么是 DP ？

$Dynamic\ Programming$ ，动态规划，指对问题**定义状态**，**划分阶段**，进行**决策性遍历**的求解问题。

~~我知道上面这段话不知道在讲啥~~

但是上面提到了三个关键词：状态，阶段，决策。相信各位在初学的时候老师一定会提到这三个词。

但是他们到底有什么用呢？如果你经过足够的训练，对于一道题，找到一个可能的**状态**，自然就可以想到转移的方式（阶段和决策）。但是这建立在你做足够题的基础上。

动态规划的转移方式通常都是**递推**，对，就是想求斐波那契数列那样。

此外记忆化搜索也可以算是一种动态规划。

所以接下来会放很多不是很难的例题出来，里面会涉及到各种各样的状态。希望这些例题将有助于各位今后的训练。

$Pt.2$ 各种 DP

2.1 背包类问题

这个大概是所有人的入门了，我默认你们都会以下的（不会待课下去看看吧）：

01背包，完全背包，多重背包，单调队列优化背包，树形背包。（较难的就不提了）。

~~感谢 $c0per\ yy$ 去年十一的 DP 课件让我刷了一周背包并且走运的在 $noip2018$ 没有因为 $D1T2$ 丢分~~

$2.1.1$ $P2744$ $[USACO5.3]$ 量取牛奶 Link

题目大意：商店里有销售 N 个桶，每个桶有一个容积 v_i 表示它可以容纳 v_i 体积的牛奶，每个桶的价格相等。FJ 希望购买最少的桶使他恰好可以测量出 Q 体积（对于两种购买桶个数相同的方案，FJ 会选择那个字典序小的方案）保证合法方案存在，请输出结果。

数据范围：$1 \le Q \le 2 \times 10^4, 1 \le N \le 100, 1 \le v_i \le 10^4$

注意到 N 非常非常小，我们可以考虑 ~~三进制枚举~~ $IDA*$，枚举答案中桶的个数，搜索所有可能的方案，对于每一种方案用背包检查其正确性。$IDA*$ 复杂度非常神奇，由于答案都不是很大，所以跑的很快，复杂度 $O(Accepted)$。

$2.1.2$ $P2347$ 砝码称重 Link

题目大意：有 $1g, 2g, 3g, 5g, 10g, 20g$ 砝码若干，请计算他们可以称出的所有可能重量。

数据范围：总重≤ 1000

~~大水题，6重循环都能过~~

直接做背包就可以了。复杂度 $O(cnt \times tot)$

$2.1.3\ JZOJ\ 19.8.18$ **提高**A**组 完全背包**

题目大意：物品 n 件，背包体积 m，物品体积 a_i 价值 b_i，每件物品无数件。求不超过容量上限的最大价值。

数据范围：$1 \le N \le 10^6, 1 \le M \le 10^{16}, a_i, b_i \le 100$

题如其名，真的是完全背包。观察数据范围后发现对于 a_i, b_i，a_i 相同的明显可以舍弃 b_i 小的，物品数量下降至 100。比较容易想到在一定范围内，贪心地选取单位体积价值大的物品，小范围内做背包。至于这个范围的大小题解给做出了相关讨论：

> 引理：给定任意 n 个整数，它们之中存在若干个整数的和为 n 的倍数。

> 定理：假设贪心的选取若干件性价比最高的物品，此外选择 x 件其他的物品是最优的情况，那么存在一种最优情况使得 $x \le val_{max}$ 。

至于证明？想想为什么。

想写的大概就只能自己造数据拿他的代码拍了。（后续可能会把题加 $Nova\ oj$ 上）

$2.1.4$ **完全背包**

题目大意：就是最普通完全背包，请输出体积 V 下使价值的一组物品选择方案。

数据范围：$1 \le N \le 100, \sum w_i \le 10000$

在转移的时候维护数组 $pre_{i,j}$ 表示当物品体积为 i 时，取的上一件物品是什么。在做完 DP 之后倒着把方案推出来就可以了。

$Pt.2.2$ **区间** DP

对区间定义状态，转移像 $f_{i,j} = \max\{f_{i,k}, f_{k,j}\}$ 的 DP 。

似乎有小部分人没有用 DP 的方法做过石子合并。所以：

2.2.1 石子合并

题目大意：N 个石子排成一排，第 i 堆石子质量为 a_i 合并相邻的两堆石子代价为 $a_i + a_{i+1}$ ，合并后原来的两堆变成一堆，求把所以石子合并成一堆的最小代价。

数据范围：$1 \le N \le 300, a_i$ 在 int 范围内。

定义状态 $f_{i,j}$ 表示把第 i 堆到第 j 堆石子合成一堆的最小代价。

初始化 $f_{i,i} = a_i$ ，转移：$f_{i,j} = \min\{f_{i,k} + f_{k+1,j} + sum_{i,k} + sum_{k+1,j}\}$

复杂度 $O(N^3)$。

核心代码长这个样子：

```
for(int i=1;i<=n;i++)f[i][i]=a[i];

for(int len=2;len<=n;len++){
    for(int i=1,j;i+len-1<=n;i++){
        j=i+len-1;
        for(int k=i;k<j;k++){
            cmin(f[i][j],f[i][k]+sum[i][k]+f[k+1][j]+sum[k+1][j]);
        }
    }
}
```

2.2.2 矩阵乘法(例题，尚未找到出处)

题目大意：给定一行 N 个矩阵，给他们添加括号，求他们进行乘法最后变成一个矩阵的最小次数。
我们认为两个矩阵 $A \times B$ 的乘法次数为 $m \times n \times p$ 次（m 行 n 列和 n 行 p 列）。
数据范围：$1 \le N \le 300$ ，数据保证任意两个相邻的矩阵都是可乘的。

定义 $f_{i,j}$ 表示 $[i, j]$ 内的所有矩阵变成一个矩阵的最小代价。

转移： $f_{i,j} = \max\{f_{i,k} + f_{k+1,j} + a_i \times a_k \times a_j\}$

2.2.3 $BZOJ1055$ $[HAOI2008]$ 玩具取名 [Link]

题目大意：某人的名字是 W, I, N, G 四个字母之一，现在可以把一个字母映射成某两个字母组成的字符串。给出这些映射和最终的字符串，求原始的字母可能是哪个。

解释：例如映射为 $W \Rightarrow II, I \Rightarrow WW, N \Rightarrow WW, G \Rightarrow IG$，最终字符串为 $IIII$，原始的字母可能为 I, N。

数据范围：最终字符串的长度≤ 200，每个字符的映射≤ 16

~~爆搜！~~，很有道理但是复杂度不对劲啊。

发现这个还原的过程是在把字符串分段，对与某个可行的分割方法，画出来必然是一棵树的样子。我们需要做的就是从树的叶子节点往根走。每次可以从某段字符串构成的节点走到上一层的某个字母的节点，相邻的两个节点可以继续向上跳。定义状态 $f_{i,j,ch}$ 表示 $[i,j]$ 这一段能不能还原成字母 ch。

转移： $f_{i,j,ch} = f_{i,k,s_1} \& f_{k+1,j,s_2} \quad [ch \Rightarrow s_1 s_2]$

2.2.4 $JZOJ$ 扭动的树

题目大意：有一棵 key 为键值 val 为权值 n 个点的二叉查找树，定义某个节点的 sum 值为它的子树内的 val 的和。告诉你 n 个节点的 key 值和 val 值，求满足树上任意一条边两个端点 key 值最大公约数不为 1 时，树上所有节点的 sum 值的和最大是多少。

数据范围：$1 \leq N \leq 300, 1 \leq key_i \leq 10^{18}, 1 \leq val_i \leq 10^6$

一棵二叉搜索树的中序遍历就是所有键值排序后的结果。试着排序后的 n 个元素进行合并，可以使用类似区间DP的思路，定义状态 $f_{i,j,k}$ 表示区间 $[i,j]$ 合并成一棵子树根为 k 的最大答案，转移类似区间DP，复杂度 $O(n^4)$。

初始化 $f_{i,i,k} = a_i \quad i\ can\ be\ k's\ son$ 转移大概是这样：

$$f_{i,j,k} = \max\{f_{i,p-1,rt_1} + f_{p+1,j,rt_2} + sum_{i,j}\} \quad [rt_1, rt_2\ can\ be\ k's\ son]$$

对这个思路进行优化：区间 $[i,j]$ 以 k 为根的情况最终一定可以由 $[i,k-1]$ 和 $[k+1,j]$ 得到。所以我们试着把上述DP的第三维改进一下，定义 $f_{i,j,0/1}$ 表示区间 $[i,j]$ 以 $i-1$ 或 $j+1$ 为根的答案，转移依然是类似区间DP的，枚举根 k，复杂度 $O(N^3)$。

转移就变成了这样：

$$f_{i,j,0} = \max\{f_{i,k-1,1} + f_{k+1,j,0} + sum_{i,j}\} \quad [i-1\ can\ be\ k's\ son]$$

$$f_{i,j,1} = \max\{f_{i,k-1,1} + f_{k+1,j,0} + sum_{i,j}\} \quad [j+1\ can\ be\ k's\ son]$$

代码在我和 $15owzly1$ 的博客里有，想做的可以课后来要题面。

2.3 图上的 DP

~~各位应该都做过过没有上司的舞会这道题~~

但是经过我小范围的调查，不少人都把这个东西忘了 ~~WTF!!!~~。要知道，在去年的比赛中 $D2T3$ 有部分分是可以写树形 DP 的，~~但是当时的我那道题一个字都没动，因为我忘了怎么写~~。

先来 ~~回顾~~ 重做一下这道题

(1) 树形 DP

这部分有些~~小~~重要，嗯，就这样。

$2.3.1$ 没有上司的舞会

题目大意：一棵带点权(v)的树，现在要在树上选择一些点，遵循以下的规则：如果选择了一个点，就不能选择它的儿子和父亲（相邻的节点）。求一个是选出来的点的权值之和最大的方案。

数据范围：$1 \le N \le 6000, -128 \le v_i \le 127$

~~你们需要把这道题非常快的切掉~~

~~其实就是求树上的一个最大权独立集。（不知道去找图论讲课人）~~

定义 $f_{i,0/1}$ 表示在以 i 为根的子树中，选择了 i 点的最大权值和。

转移：

$f_{i,0} = \sum \max(f_{j,0}, f_{j,1}) \quad [j\ is\ i's\ son]$

$f_{i,1} = \sum f_{j,0} \quad [j\ is\ i's\ son]$

$2.3.2$ 树的直径 I

题目大意：给定一棵 N 个点的树，每条边的长度为 1。求树的直径的长度。

数据范围：$1 \le N \le 10^5$

两遍 dfs，从第一次求得的最远点出发，走到的最远点和它组成直径的端点。另外直径可能有多条。

定义 1 为整棵树的根。设 f_i 表示在以节点 i 为根的子树中，从 i 出发向下最大的深度， g_i 表示在以节点 i 为根的子树中，从 i 出发向下次大的深度。初始化 $f_i = 1, g_i = 0$。转移：

$f_i = \max\{f_j + 1\} \quad [f_i < f_j + 1 \; and \; j \; is \; i's \; son]$

$g_i = \max\{g_j + 1\} \quad [f_i > f_j + 1 \; and \; j \; is \; i's \; son]$

在做到每一个节点的时候更新答案就可以了。

$2.3.2$ 树的直径 II

题目大意：给定一棵 N 个点的树，每条边有一个长度 $w_{x,y}$。求树的直径的长度。

同上：

$f_i = \max\{f_j + w_{i,j}\} \quad [f_i < f_j + w_{i,j} \; and \; j \; is \; i's \; son]$

$g_i = \max\{g_j + w_{i,j}\} \quad [f_i > f_j + w_{i,j} \; and \; j \; is \; i's \; son]$

可以再思考一下怎么记录方案。

$2.3.3$ $P2585$ $[ZJOI2006]$ 三色二叉树 [Link]

题目大意：一棵 N 个节点的二叉树，每个点只能染红、蓝、绿三种颜色，每个点不能和它相邻的节点同色，每个点的儿子也不能同色。求最多能染多少个绿色节点。

数据范围：$1 \le N \le 5 \times 10^5$

原题的输入非常毒瘤，但是问题不大。我只讲 DP 的部分。

比较简单的定义 $f_{i,0/1/2}$ 表示 i 号节点为颜色 $0/1/2$ 时子树内最多有多少个绿色节点。

转移无非就是枚举一下所有的可能性。

如：$f_{i,0}=\max\{f_{son_1,1}+f_{son_2,2}\}$

发现有几种相同的转移，状态可以改成 $f_{i,0/1}$ 表示 i 号节点是不是绿色点，转移类似上面。

$(2)\ DAG$ 上 DP

图上 DP，就是在图上 DP。

其实所有的 DP 的状态和转移都可以看成 DAG（如果不是 DAG 就有后效性），转移的过程中遍历状态的过程就像拓扑排序（~~什么不会？图论讲课人呢？~~）。

这里放几道非常裸的在图上点之间转移的题。

~~你看这个最短路也有点像 DP~~

$2.3.4\ P3183[HAOI2016]$ ~~你们大多数人都做过的~~ 食物链

题目大意：求一张 N 个点 M 条边的 DAG 上有多少条完整的链（头尾不能再延伸）。

数据范围：$1\le N\le 10^5, 1\le M\le M\le 2\times 10^5$

$f_i=1\quad[out_i=0]$

$f_i=\sum f_j$

$2.3.5\ P1137$ 旅行计划 [Link]()

题目大意：一个 N 个点 M 条边的 DAG，求以每个点为终点的路径最长是多少。

数据范围：$1\le N\le 10^5, 1\le M\le 2\times 10^5$

建反图，拓扑排序式的转移。

$f_i=\max\{f_j+1\}$

2.4 状态压缩 DP

对于小数据，定义非常非常暴力的状态，做非常非常暴力的转移。

2.4.1 $P2622$ 关灯问题II Link

题目大意：现有n盏灯，以及m个按钮。每个按钮可以同时控制这n盏灯——按下了第i个按钮，对于所有的灯都有一个效果。按下i按钮对于第j盏灯，是下面3中效果之一：如果a[i][j]为1，那么当这盏灯开了的时候，把它关上，否则不管；如果为-1的话，如果这盏灯是关的，那么把它打开，否则也不管；如果是0，无论这灯是否开，都不管。

现在这些灯都是开的，给出所有开关对所有灯的控制效果，求问最少要按几下按钮才能全部关掉。

数据范围：$1\le N\le 10,1\le m\le 100$

不好意思题面没有整理。

定义 f_s 表示 N 盏灯状态为 s 时的最少操作次数。0 表示亮，1 表示灭，初始化 $f_0=0$ 。

转移：$f_{now}=\min\{f_{pre}+1\}\quad [x\ option\ can\ change\ pre\ to\ now]$

2.4.2 $P1879\ [USACO06NOV]$ 玉米田 Link

题目大意：在一块 $N\times M$ 的田地里，每块土地都是被标记为可以选择或者不能选择，选择一些可以选择的土地，要求选择的土地不能有相邻的，求方案数。

数据范围：$1\le N,M\le 12$

定义 $f_{i,s}$ 表示第 i 行选择的土地为 s 状态时的合法方案数。

预处理出所有合法的状态，两行间的状态通过与运算判断是否合法。

2.5 期望和概率 DP

这部分内容主要摘自一个课件。

2.5.1 $BZOJ$ 1419 纸牌？

题目大意：桌面上有 A 张红牌与 B 张黑牌，随机打乱放置。现在开始一张一张的翻盘，翻到红牌加一分，黑牌一分，求在最优策略的情况下的期望得分。
数据范围：$A, B \le 5000$

设 $f_{i,j}$ 表示桌上还剩下 i 张黑牌，j 张红牌的期望得分。

最佳利益，分类讨论一下
当 $i = 0$ 时 $f_{i,j} = j$
当 $j = 0$ 时 $f_{i,j} = 0$
否则的话 $f_{i,j} = \max(0, \frac{i}{i+j}(1 + f_{i-1,j}) + \frac{j}{i+j}(-1 + f_{i,j-1}))$

2.5.2 $JZOJ$ [纪中提高A组 19.8.15 测试] 投票

题目大意：有 n 个同学投票，每个人有一个投赞成票的概率 p_i。现在要从中选出 k 个同学，使得这 k 个人投票的平票的概率最大。

数据范围：$1 \le N, K \le 2000$

考场上自己生成数据找规律发现一定是对所有人的 p_i 排序后头尾各取几个人（可以不取），~~做 $O(N^2)$-DP 拿到了 70 分~~。这个性质的正确性具体来说就是假设已经确定了一些人，再选一个人结果是关于选的这个人的 p 的一个一次函数（其他部分已经确定，为常数）。具体推导可以看：here。利用这条性质，我们只需要对前缀后缀各 $O(N^2)$ 做一遍DP算出概率再 $O(N)$ 枚举前面选几个人即可。复杂度 $O(N^2)$
。

2.6 计数类 DP

如果看到那种 $\mod 998244353$ 这样的求方案数的题，考虑数学做法，不行就考虑 DP 。

定义某个状态 f_i 为 i 状态的所有可能性。

2.7 数位 DP

你看到题面要求统计一些满足条件的数字，感觉好像可做。

但是当你发现数据范围超级大时，~~不禁鼻子酸酸的~~。

$2.7.1$ hdu 2089 不要62

题目大意：求区间 $[N,M]$ 内不含 62 连号的个数。

数据范围：$0 \le N,M \le 10^6$

定义 $f_{i,x}$ 表示一个 i 位数以 x 结尾合法的方案数。转移是判断 6 不能到 2。

$Pt.3$ DP 的优化

算下时间复杂度，$Boom!$，这时候就要考虑考虑优化了。

3.1 矩阵加速递推

矩阵乘法

指数的递推：

$3.1.1$ $JZOJ$ [纪中提高A组 19.8.7 测试] 小L的序列

题目大意：

给出$f_1,f_2,...f_k$和$b_1,b_2,...b_k$。

$f_i=(f_{i-1}^{b_1}\cdot f_{i-2}^{b_2}\cdots f_{i-k}^{b_k}) mod\, p\ (i\ge k)$

求f_n。

数据范围：$1 \le N \le 4 \cdot 10^6, 1 \le K \le 200, P = 998244353, 1 \le f_i, b_i \le P$

PS：以前在CF里见到过一个$k = 3$的情况，当时再考场上推到第8项发现每一项的指数都可以递推，当时搞不明白为什么，今天想的时候就清楚了。

最终的答案一定是每个f_i的若干次方的乘积组成的，考虑计算每个f_i最终的指数。我们前k项中的某一项在第i个位置的指数是dp_i，两个f_i相乘是加法操作，f_x的次方是乘法操作，那么就有$dp_i = \Sigma_{j=1}^{j \le k} dp_{i-j} \cdot b_j$，根据费马小定理，我们需要对$dp$模$P-1$。

这个转移可以用矩阵乘法优化：

$$[\,dp_i\ dp_{i+1}\ dp_{i+2} \ldots dp_{i+k-1}\,] \cdot \begin{bmatrix} 0\,0\,0 \ldots\ 0\ b_k \\ 1\,0\,0 \ldots\ 0\ b_{k-1} \\ 0\,1\,0 \ldots\ 0\ b_{k-2} \\ \ldots \\ 0\,0\,0 \ldots\ 1\ b_1 \end{bmatrix} = [\,dp_{i+1}\ dp_{i+2}\ dp_{i+3} \ldots dp_{i+k}\,]$$

对于前k项中的每个位置，只要分开做就可以得到每个位置最后的指数，当然，把每个位置的初始矩阵拼起来做一次也是可以的。

$[trick]$ $2.1.3$ $JZOJ$ $19.8.18$ **提高A组 完全背包**

题目大意：物品 n 件，背包体积 m，物品体积 a_i 价值 b_i，每件物品无数件。求不超过容量上限的最大价值。

数据范围：$1 \le N \le 10^6, 1 \le M \le 10^{16}, a_i, b_i \le 100$

在前面的解法之外还存在一种矩阵优化普通的 dp 的做法。

通常的矩阵加速递推是形如 $f_i = \sum_{j=i-k}^{i-1} c_{i-j} \cdot f_j$ 的形式，用普通的矩乘就能优化。对于 $\min / \max$ 一类的操作，他们和加法具有类似的性质。如果定义一种矩阵的取 $\max$ 运算（$A\ \max\ B = C, c_{i,j} = \max\{max(a_{i,k}, b_{k,j})\}$），这种运算满足结合律，也可以用矩乘快速幂优化。

代码请转至：https://www.cnblogs.com/15owzLy1-yiylcy/protected/p/11379610.html

鸣谢 $15owzly1$。

3.2 数据结构

针对一些转移的特点，比如 $f_i = \max\{f_j + 1\} \quad [a_i \le a_j]$，可以选择数据结构来维护转移。

比如上面的式子就可以用线段树优化。多重背包可以用单调队列优化。

单调栈、单调队列、线段树都是我们常用的数据结构。~~（特殊的时候还会用平衡树）~~

$Pt.4$ 杂题

一些我无法分类的题目。

之后贴。

$Pt.5$ 没来得及写题解但是值得一做的题

之后贴。