

dp题目选讲

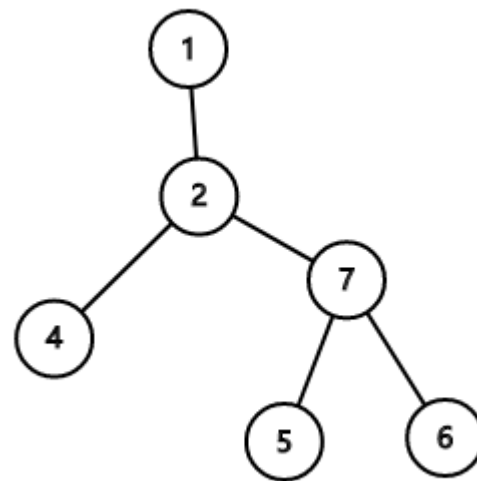
GAREN 2019.05.12

P2014 选课

- ▶ 在大学里每个学生，为了达到一定的学分，必须从很多课程里选择一些课程来学习，在课程里有些课程必须在某些课程之前学习，如高等数学总是在其它课程之前学习。现在有 N 门功课，每门课有个学分，每门课有一门或没有直接先修课（若课程 a 是课程 b 的先修课即只有学完了课程 a ，才能学习课程 b ）。一个学生要从这些课程里选择 M 门课程学习，问他能获得的最大学分是多少？

分析

- ▶ 这道题就是赤裸裸的树形背包模板题。
- ▶ 先给出样例的依赖关系，方便解题。



分析

- ▶ 看到先修课，就想到了DAG。看到每科最多只有一门先修课，就想到森林。
- ▶ 注意：给你的依赖关系不一定是棵树，可能有多个节点。显然不能暴力套dp。
- ▶ 其实可以很简单地解决：设一个超级源点 s ，把那些根节点都连向 s 即可。
- ▶ 直接揭晓状态设计： $dp[u][j]$ 为以 u 为根的子树中，选了 j 个节点的最大价值。
- ▶ 然后有显然的转移方程： $dp[u][j] = \max(dp[u][j - k] + dp[v][k])$
- ▶ 实现的时候要注意： j 要倒序枚举。同时要保证转移的状态是合法的。
- ▶ 因为上面的状态已经滚过了。本来是 $dp[u][i][j]$ 表示以 u 为根的子树中，前 i 个点选了 j 个的最大价值。而这道题是01背包，所以逆序枚举 j 。
- ▶ 初始状态是 $dp[u][1] = val[u]$ 。最终答案是 $dp[0][m + 1]$ 。

P4644 [Usaco2005 Dec]Cleaning Shifts

清理牛棚

- ▶ 约翰的奶牛们从小娇生惯养，她们无法容忍牛棚里的任何脏东西。约翰发现，如果要使这群有洁癖的奶牛满意，他不得不雇佣她们中的一些来清扫牛棚。约翰的奶牛中有 $N (1 \leq N \leq 10000)$ 头愿意通过清扫牛棚来挣一些零花钱。由于在某个时段中奶牛们会在牛棚里随时随地地乱扔垃圾，自然地，她们要求在这段时间里，无论什么时候至少要有一头奶牛正在打扫。需要打扫的时段从某一天的第 M 秒开始，到第 E 秒结束 ($0 \leq M \leq E \leq 86399$)。注意这里的秒是指时间段而不是时间点，也就是说，每天需要打扫的总时间是 $E - M + 1$ 秒。约翰已经从每头牛那里得到了她们愿意接受的工作计划：对于某一头牛，她每天都愿意在第 T_1, T_2 秒的时间段内工作 ($M \leq T_i \leq E$)，所要求的报酬是 S 美元 ($0 \leq S \leq 500000$)。与需打扫时段的描述一样，如果一头奶牛愿意工作的时段是每天的第 10_20 秒，那她总共工作的时间是 11 秒，而不是 10 秒。约翰一旦决定雇佣某一头奶牛，就必须付给她全额的工资，而不能只让她工作一段时间，然后再按这段时间在她愿意工作的总时间中所占的百分比来决定她的工资。现在请你帮约翰决定该雇佣哪些奶牛以保持牛棚的清洁，当然，在能让奶牛们满意的前提下，约翰希望使总花费尽量小。

分析

- ▶ 这道题就是之前口胡过的带权线段覆盖求最小值。
- ▶ 显然无法贪心。反例是很多的。
- ▶ 无法贪心自然就想到dp。设 $dp[i]$ 为覆盖 $[s, i]$ 的最小权值。（ s 为起点）
- ▶ 但是对一堆线段无法直接dp，不满足无后效性。所以按右端点排下序。
- ▶ 状态转移方程为 $dp[r] = \min_{i=l-1}^r dp[i] + w$ 。
- ▶ 这样就能算出答案。但是显然 $O(n^2)$ 的复杂度，跑不过这道题。
- ▶ 考虑我们转移的时候要找出 $[l-1, r]$ 中最小的 dp 值，我们可以用数据结构维护。比如线段树。
- ▶ 所以有 $O(n \log n)$ 的复杂度。

P1156 垃圾陷阱

- ▶ 卡门——农夫约翰极其珍视的一条Holsteins奶牛——已经落到了“垃圾井”中。“垃圾井”是农夫们扔垃圾的地方，它的深度为 D ($2 \leq D \leq 100$)英尺。
- ▶ 卡门想把垃圾堆起来，等到堆得与井同样高时，她就能逃出井外了。另外，卡门可以通过吃一些垃圾来维持自己的生命。
- ▶ 每个垃圾都可以用来吃或堆放，并且堆放垃圾不用花费卡门的时间。
- ▶ 假设卡门预先知道了每个垃圾扔下的时间 t ($0 < t \leq 1000$)，以及每个垃圾堆放的高度 h ($1 \leq h \leq 25$)和吃进该垃圾能维持生命的时间 f ($1 \leq f \leq 30$)，要求出卡门最早能逃出井外的时间，假设卡门当前体内有足够持续10小时的能量，如果卡门10小时内没有进食，卡门就将饿死。

分析

- ▶ 这是一道考你状态设计以及实现细节的dp题。
- ▶ 面对每一件垃圾，都有吃或堆起两种选择，与背包的选与不选有一点点类似。
- ▶ 我们可以类似背包的dp状态，定义 $dp[i][j]$ 为用前 i 件垃圾堆起高度 j 的最大生命。
- ▶ 由于给出的垃圾是乱序的，所以我们按时间排序后才能用来dp。
- ▶ 这道题与背包的转移不同的是：不存在不选的情况，会把垃圾吃了然后+1s。
- ▶ $dp[i][j] = \max(dp[i-1][j] + f[i] - \text{delta}, dp[i-1][j - h[i]] - \text{delta})$
- ▶ 注意：上面的状态转移方程一定要满足条件才能转移，否则就不转移。
- ▶ 如果有解的话，一定存在某个 $dp[i][m]$ 被成功转移到。
- ▶ 如果无解的话就有两种方法可以解决：一种是贪心地把垃圾全吃了，看能够活多久；另一种是找到合法且最大的 $dp[i][j] + t[i]$ 。

P1941 飞扬的小鸟

- ▶ Flappy Bird是一款风靡一时的休闲手机游戏。玩家需要不断控制点击手机屏幕的频率来调节小鸟的飞行高度，让小鸟顺利通过画面右方的管道缝隙。如果小鸟一不小心撞到了水管或者掉在地上的话，便宣告失败。
- ▶ 为了简化问题，我们对游戏规则进行了简化和改编：
- ▶ 游戏界面是一个长为 n ，高为 m 的二维平面，其中有 k 个管道（忽略管道的宽度）。
- ▶ 小鸟始终在游戏界面内移动。小鸟从游戏界面最左边任意整数高度位置出发，到达游戏界面最右边时，游戏完成。
- ▶ 小鸟每个单位时间沿横坐标方向右移的距离为 1，竖直移动的距离由玩家控制。如果点击屏幕，小鸟就会上升一定高度 X ，每个单位时间可以点击多次，效果叠加；如果不点击屏幕，小鸟就会下降一定高度 Y 。小鸟位于横坐标方向不同位置时，上升的高度 X 和下降的高度 Y 可能互不相同。
- ▶ 小鸟高度等于 0 或者小鸟碰到管道时，游戏失败。小鸟高度为 m 时，无法再上升。
- ▶ 现在,请你判断是否可以完成游戏。如果可以，输出最少点击屏幕数；否则，输出小鸟最多可以通过多少个管道缝隙。

分析

- ▶ ~~我小时候好像最多玩到100+~~
- ▶ 如果抱着骗分的心态来的话显然直接bfs乱搞搜索就ok了。个人最好成绩65pts。
- ▶ 其实这道题正解是dp，bfs要加上千奇百怪的剪枝和玄学优化才能过吧。
- ▶ 考虑小鸟只有两种决策，要不往上飞，要不往下掉。
- ▶ 往上飞能飞无限次（除非高度限制），而往下却只能掉一次，不禁让人想到完全背包和01背包。
- ▶ 所以我们直接设出 $dp[i][j]$ 表示小鸟跳到 (i,j) 时最少的点击数，然后考虑用背包的转移方式实现转移。

分析

- ▶ 枚举当前点 (i, j) 后，考虑用填表法解决掉落：如果合法就直接从前面的转移来即可。
- ▶ 这道题的难点就在于向上飞的转移。
- ▶ 最朴素的就是枚举我们跳几次跳上去，进行转移。代码上就多了一层 k 的循环。
- ▶ 这种方法能拿到75pts，考虑怎么优化掉这层 k 的循环。
- ▶ 这个时候我们用到完全背包优化的一个技巧：既从 $dp[i - 1][j - jump[i - 1]]$ 转移，又从 $dp[i][j - jump[i - 1]]$ 转移来。
- ▶ 但是这就出现了问题： $dp[i][j - jump[i - 1]]$ 所在的点可能非法，难道能转移吗？
- ▶ 其实只需要在更新的最后，把这些非法点继续改为非法值就可以了。在中途我们利用这些点优化转移，最后再变回原来的水管。

分析

- ▶ 有解的话就从所有的 $dp[n][i]$ 中找最小。无解的话就暴力从后往前遍历，找到横坐标最大的被转移的点，算下水管和即可。
- ▶ 有一个奇怪的点：要先完全背包后01背包，否则会哇两个点。

后面的话

- ▶ 这个月考之后我可能就要撒手人寰了。之后的就要你们自己安排了。
- ▶ 然后我这边还有一个GitHub项目：信息学知识系统。在我的GitHub账号里面有。
- ▶ 去年这个时候就在里面写了很多东西，但是还是不全面。今年靠你们去补齐了。
- ▶ NOIP少于180天了，rp++

再见