~~树型背包~~

~~一般的树型背包~~

树形背包

一般的树型背包

一般树上背包dp统计的是i子树内选j个节点的信息，即$dp(i,j)$

转移时，枚举背包大小

$$i\in[0,sz_u],\ j\in[0,sz_v],\quad dp(u,i)+dp(v,j)\to re(i+j)$$

这里sz_u表示的是已经处理完的u的子树大小，动态更新。

复杂度$O(n^2)$而不是$O(n^3)$

有上界的树上背包

有些题目的背包容量不仅受子树大小的限制，还受一个参数K的限制，在转移时

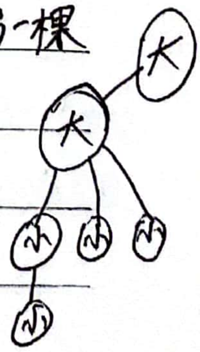
$$i\in[0,\min(sz_u,K)],\ j\in[0,\min(sz_v,K)],\quad dp_{u,i}+dp_{v,j}\to re_{i+j}$$

这样做的复杂度是$O(nK)$

证明如下：

我们把每个节点对应的子树根据是否大于K分别大子树和小子树。

每条边都对应一次dp转移，把一棵子树的信息合并到另一棵子树中去。



①对于小-小合并，我们观察那些极大小树，每棵树当作普通dp来算复杂度，有$\sum sz^2_{极大小树}$，最坏情况是所有sz取到K，复杂度为$\frac{n}{K}\cdot K^2=nK$

②对于大—小合并，最多只有n条这样边，且这些小子树大小和$\le n$，转移的第一层循环算k次，第二层循环次数之和为$\sum sz_{\text{极小树}} \le n$，复杂度不超过$nK$.

③对于大—大合并，大于K的块和大于K的块之间合并最多有$\frac{n}{K}$次，复杂度不超过$\frac{n}{K}\cdot K^2 = nK$.

带点权的树上背包

树上每个节点都有体积w_i和价值v_i，并且物品的选取有依赖关系，只有子树根节点被选才能选子树内的物品。

解法：后序遍历整棵树，记录sz_i，在后序dfs序上做dp。

设$dp_{i,j}$表示考虑了前i个物品，背包容量为j能获得的最大价值。

有转移方程（选/不选根节点）：

$$dp(i,j) = \max \begin{cases} dp(i-1, j-w_{dfn_i}) + v_{dfn_i}, & \text{if } j \ge w_{dfn_i} \\ dp(i-sz_{dfn_i}, j) \end{cases}$$

其中，dfn_i表示后序遍历的第i个节点，

sz_u表示u的子树大小、后序遍历dfs序中，

dfn_i的子树节点刚好是$dfn_{[i-sz_{dfn_i}, i]}$，即前sz个。

状态总数$n\times W$个，转移$O(1)$

复杂度$O(nW)$

注意这种题不能用sz_u限制转移次数和背包容量，无法用之前的解法。另外，此解法只能求出整棵树的背包信息，无法对某棵子树求出容量为j的限制下能获得的最大价值。

Slope trick 优化dp

定义 ~~等~~ 可斜函数 (slope-trick-able):

① 该函数连续

② 可分割成许多段一次函数段(线段),且每段斜率为整数。

③ 该函数是凸/凹的,即从左向右,每段斜率在递增/减(或全部递减)

定义斜率改变点 x_{change}:

就是可斜函数的分段点,且 x_c 左边斜率小于/大于 x_c 右边斜率。

更进一步,规定 $f(x)$ 在 x_c 斜率刚好改变 1。如果 $f(x)$ 在某个分段点处斜率改变不止 1,那么我们认为这存在多个斜率改变点。

有了上两个定义,我们就能用 $f(x)$ 的最右段一次函数和一个斜率改变点的多重集合来表示 $f(x)$ 了,例如

$$f(x)=\begin{cases}2, & x<2\\ x, & 2\le x<8\\ 4x-24, & 8\le x\end{cases}$$

用 $[y=4x-24,\ S=\{2,8,8,8\}]$ 来表示 $f(x)$,通过以下方式来重构 $f(x)$:

~~$4x-24=3x+b$ 在 $x=8$ 处成立 $\Rightarrow$ $b=-16$,即 $y=3x-16$ 是第二段函数~~

3个连续的8,代表斜率减3,那设第二段函数为 $y=x+b$

有 $x+b=4x-24\ (x=8)\ \Rightarrow\ b=0$,即第二段函数为 $y=x$。

同理,$c=x\ (x=2)\ \Rightarrow\ c=2$,即第一段为 $y=2$。

性质　　　　　　　　　　　　全是凸/凹函数

两个可斜函数的表示是可合并的,若 $f(x), g(x)$ 是可斜的,那么 $h(x)=f(x)+g(x)$ 也是可斜函数,并且 $h(x)$ 的最右一次函数等于 $f(x)$ 最右段加上 $g(x)$ 最右段,$h(x)$ 的斜率改变点刚好 $S_h=S_f\cup S_g$。

Slope trick 的本质就是灵活的修改最右一次函数和斜率改变点。

应用

一、CF713C：给一个数组 $a_1\sim a_n$，每次操作选一个 a_i 加1或减小1，问最小步数，使得 $a_1\sim a_n$ 递增.

解：对所以 $a_i \leftarrow a_i - i$，原问题转为让 $a_1\sim a_n$ 非递减.

定义 $f_{i,j}$ 表示 $a_1\le\cdots\le a_i = j$ 的最小步数，$g_{i,j}$ 为 $a_1\le\cdots\le a_i\le j$ 的最少步数，

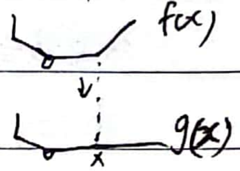
有转移 $f_{i,j} = g_{i-1,j} + |a_i - j|$，$g_{i,j} = \min\limits_{k\le j} f_{i,k}$.

使用可斜函数 $f_i(x)$ 和 $g_i(x)$ 代替上式：

$$f_i(x) = g_{i-1}(x)+|x-a_i|,\quad g_i(x)=\min_{j\le x} f_i(j)$$

可归纳性的证明 $f_i(x)$ 和 $g_i(x)$ 是可斜函数，且 $f_i(x)$ 的最右斜率为1，$g_i(x)$ 的最右斜率为0，$g_i(x)$ 相当于把 $f_i(x)$ 的最右段"推平"。

所以，维护 $f(x)$ 的最右一次函数和最右 x_{change} 即可。

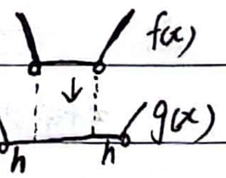


二、NOI2018 safety：给定 $a_1\sim a_n$，每次选 a_i 加1/减1，使 $|a_i - a_{i+1}|\le h$，问最小步数，

解：同上题，$f_i(x)=g_{i-1}(x)+|x-a_i|$，$g_i(x)=\min\limits_{j=x-h}^{x+h} f_i(j)$.

可以想象，$g(x)$ 相当于把 $f(x)$ 的"谷底"向左右各拉长 h。

对于左右斜坡各用个堆来维护 x_{change}，同时维护 $f^R(x)$，$g^R(x)$.



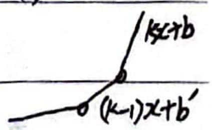
① 对于拉长 h，用两变量记录 S_f 的改变量，即懒标记，

同时，$g_i^R(x)=f_i^R(x-h)$.

② 对于 $f_i(x)=g_{i-1}(x)+|x-a_i|$，判断左右堆的"谷底" x_c，对 S_f 进行改变，

可以发现，$S_f\leftarrow S_f\cup\{a_i,a_i\}$，对左右优先队列的 pop 最多一次，

同时，$f_i^R(x)=g_{i-1}^R(x)+x-a_i$.



③ 最后还原每一段时，从右往左找 x_c，新的一段 $b'=x_c+b$，$k'=k-1$.

△ 事实上，我们可以不维护最右一次函数，转而维护"谷底"一次函数 $(k=0)$，

这样做同样能还原全函数，并且往往我们只关注峰值，不需要还原全函数了，

直接输出答案。