dp 优化小技巧全家桶

lam1789

7.28

单调队列优化dp

Watching Fireworks is Fun (CF372C)

一个城镇有 n 个区域，从左到右编号为 $1 \sim n$，个区域之间距离 1 个单位距离。
有 m 个烟火要放，给定放的地点 a_i，时间 t_i，如果你当时在区域 x，那么你可以获得 $b_i - |a_i - x|$ 的开心值。
你每个单位时间可以移动不超过 d 个单位距离。
你的初始位置是任意的（初始时刻为 1），求你通过移动能获取到的最大的开心值。
$1 <= n <= 150000$; $1 <= m <= 300$; $1 <= d <= n$。

Watching Fireworks is Fun (CF372C)

设 $f_{i,j}$ 表示放第 i 个烟花时，正处于位置 j 时可获得的最大快乐值。
有转移方程：$f_{i,j} = (\max_{k=j-d}^{j+d} f_{i-1,k}) + b_i + |a_i - j|$。
从 1 到 n 枚举 j，发现括号里的东西是一个滑动窗口问题，用单调队列可做到 $\Theta(n)$ 求解。
时间复杂度 $\Theta(nm)$。

斜率优化

HNOI2008 玩具装箱(luogu P3195)

有 n 个玩具，第 i 个玩具价值为 c_i，将 n 个玩具排成一排，分成若干段。对于一段 $[l, r]$，它的代价是 $(r - l + \sum_{i=l}^{r} c_i - L)^2$。其中 L 是一个常数。

$n \leq 5 \times 10^4, 1 \leq L, c \leq 10^7$。

HNOI2008 玩具装箱(luogu P3195)

设 f_i 表示前 i 个物品的最小代价，列出方程：
$f_i = \min_{j=1}^{i-1}(f_j + (i - j - 1 - L + \sum_{k=j+1}^{i} c_i))$
进行前缀和优化，令 $s_n = \sum_{i=1}^{n} c_i$，有：
$f_i = \min_{j=1}^{i-1}(f_j + (i - j - 1 - L + s_i - s_j)^2)$
时间复杂度 $\Theta(n^2)$，无法通过。

HNOI2008 玩具装箱(luogu P3195)

考虑令 $g_i = i + s_i$，$k = L + 1$有：
$f_i = \min_{j=1}^{i=1}(f_j + (g_i - g_j - k)^2)$。
化简：$f_i - (g_i - k)^2 = \min_{j=1}^{i-1}((f_j + g_j^2) - 2g_j(g_i - k))$。
考虑将与与 j 相关的看作一个点，而将 i 相关的看作一条直线。
具体的，设点为 (x_j, y_j)，直线为 $y = k_i x + b_i$。具体的：
$x_i = g_j, y_j = f_j + g_j^2, k_i = 2(g_i - k), b_i = f_i - (g_i - k)^2$。
由于 $b = y - kx$，所以可以将问题看作：
给定 k，选择一个点 (x, y)，使斜率为 k 的直线经过，最小化截距。

HNOI2008 玩具装箱(luogu P3195)

怎么找这一个点？考虑先让截距从 $-inf$ 开始将直线向上平移，碰到的第一个点即为要选的点。发现这个点一定在点集的下凸壳上，且这个点要选，当且仅当其与其左边点的连线的斜率 $\leq k$，其与其右边点的连线的斜率 $\geq k$。
那么如何维护凸包呢？我们从 1 到 n 依次加入 (x_i, y_i)，因为 $x_i = g_j = i + s_i$ 是单调递增的，所以只需要用单调栈维护凸包就好了。
时间复杂度 $\Theta(n \log n)$。

平衡树优化 dp & CDQ 分治优化 dp

HNOI2008 玩具装箱改

有 n 个玩具，第 i 个玩具价值为 c_i，将 n 个玩具排成一排，分成若干段。对于一段 $[l, r]$，它的代价是 $(r - l + \sum_{i=l}^{r} c_i - L)^2$。其中 L 是一个常数。
$n \le 5 \times 10^4, -10^7 \le L, c \le 10^7$。

HNOI2008 玩具装箱改

与原题唯一的区别是 c_i 可以为负，导致 x_j 不再单调递增，无法使用单调栈维护凸包形态。现在怎么办呢？
一种想法是使用平衡树维护，加入一个点时，在其左边与右边各删除一个连续段的点。但是常数巨大且难写！

CDQ 分治

引入算法：CDQ 分治。
CDQ 分治的核心思想是将序列劈成两半，先计算左半边，右半边内的答案，再计算左半边对右半边的影响。
这样说有点抽象，看个例题：

三维偏序(luogu P3810)

有 n 个元素，第 i 个元素有 a_i, b_i, c_i 三个属性。令 f_i 为满足 $a_j \leq a_i, b_j \leq b_i, c_j \leq c_i$ 且 $(i != j)$ 的 j 的数量。

三维偏序(luogu P3810)

考虑 CDQ 分治。将元素按 a_i 排序，然后分治。当我们处理区间 $[l, r]$ 时，我们要满足以下目标：

1. 求出所有的 $l \le j < i \le r$ 产生的贡献。
2. 将 $[l, r]$ 之间的所有元素按照 b 排序（有点类似归并排序求逆序对的思想）。

令 $mid = \lfloor \frac{l+r}{2} \rfloor$，我们先递归处理 $[l, mid]$ 与 $[mid+1, r]$。
然后与归并排序求逆序对类似，但是额外维护一个树状数组。维护两个指针 l_1, l_2，若 $b_{l_1} <= b_{l_2}$ 则将 l_1 插入排序后的新序列中，并额外在树状数组的 c_i 处加上 1。若 $b_{l_2} < b_{l_1}$，则将 l_2 插入排序后的新序列中，此时发现，树状数组中 1 到 c_{l_2} 的和即为左半边所有元素对 l_2 的贡献。
时间复杂度 $\Theta(n \log^2 n)$。

HNOI2008 玩具装箱改

考虑使用 CDQ 分治。
当我们希望求出 (l, r) 之间的 f 时，先递归处理 (l, mid)，并将 (l, mid) 的点按 x 排序并造出凸壳，用它去更新 $(mid+1, r)$ 的答案。对于其中最优决策在 (l, mid) 中的点，其均得到更新，(l, mid) 中的决策点已发挥所有作用。接下来递归处理 $(mid+1, r)$ 即可。
时间复杂度 $\Theta(n \log^2 n)$。

李超树

请看例题。

[HEOI2013] Segment(luogu P4097)

要求在平面直角坐标系下维护两个操作：
1. 在平面上加入一条线段。记第 i 条被插入的线段的标号为 i。
2. 给定一个数 k，询问与直线 $x = k$ 相交的线段中，交点纵坐标最大的线段的编号。
$1 \leq n \leq 10^5$

[HEOI2013] Segment(luogu P4097)

考虑在线段树上维护以下信息：
1.单点 x 上 y 的最大值和所属的线段。
2.懒标记(完全覆盖该区间，且在区间中点处 y 的最大值所属的线段)。
考虑向某个区间插入一条线段时，将懒标记线段拎出。比较它们在 l,r 处的取值。如果有一条在 l,r 都更大，则说明它在整个区间中都比另一条更大。将懒标记替换为它。
否则比较它们在 mid 处的取值。设 mid 处取值更大的线段是 a，另一条是 b。将懒标记变为 a，若 a 在 r 处取值比 b 小，则递归处理 $[mid+1,r]$，否则递归处理 $[l,mid]$。
查询时查询到该单点路径上所有懒标记的最大值即可。

李超树实现斜率优化

HNOI2008 玩具装箱改

有 $f_i - (g_i - k)^2 = \min_{j=1}^{i-1}((f_j + g_j^2) - 2g_j(g_i - k))$。这次令 $k_j = -2g_j, b_j = (f_j + g_j^2), x_i = (g_i - k), y_i = f_i - (g_i - k)^2$。
发现这次就是维护一堆直线集合，每次查询所有直线中 x_i 位置最大的 y 值。
这就是李超树了，用李超树维护即可。

决策单调性

Q：什么是决策单调性？
A：对于 dp 转移方程 $dp_i = \max_{j<i} dp_j + f_j$，令其最优决策点为 d_i，有 $\forall j < i, d_j < d_i$，则称转移方程有决策单调性。

决策单调性

Q：怎么判定是否具有决策单调性？
A：请学习：OI-Wiki 四边形不等式。
但这太麻烦了，考虑到我们是信息学竞赛，所以只需要打个表出来瞪一眼就好了。

决策单调性

Q：决策单调性有什么用？
A：请看题：

[JSOI2016] 灯塔 (luogu P5503)

给定正整数序列 h，对于每个 i，求出最小的非负整数 p，满足 $\forall j, h_j \leq h_i + p + \sqrt{|i-j|}$。
$1 \leq n \leq 10^5$。

[JSOI2016] 灯塔 (luogu P5503)

不妨把 $i > j$ 和 $i < j$ 两种情况分开讨论。不妨令 $j < i$，有 $ans_i + h_i = \max(h_j + \lceil\sqrt{i-j}\rceil)$。
不妨我们已经通过打表发现了有决策单调性。令 p_i 为 ans_i 的最优决策点。
考虑分治。考虑希望计算 $[l, r]$ 的 ans, p，其最优决策点在 $[L, R]$ 中。
求出 $mid = \lfloor\frac{l+r}{2}\rfloor$ 的最优决策点 MID，然后递归计算 $[l, mid-1], [L, MID]$ 与$[mid+1], [MID, R]$ 即可。
时间复杂度 $\Theta(n\log n)$。

[JSOI2016] 灯塔 (luogu P5503)

一个奇怪的做法。
依然考虑 $ans_i + h_i = \max(h_j + \lceil\sqrt{i-j}\rceil)$。发现 $\lceil\sqrt{i-j}\rceil$ 只有 $\Theta(\sqrt{n})$ 个取值。枚举它们，用 ST 表维护区间最大值即可。
时间复杂度 $\Theta(n\sqrt{n})$，可以通过。

来源应该是忘了哪个山东省集的题了

有两个长度分别为 n, m 的小写字母字符串 A, B，求出最长公共子序列。

$n \leq 10^6, m \leq 10^3$。

来源应该是忘了哪个山东省集的题了

传统做法：设 $f_{i,j}$ 为 A 的前 i 位与 B 的前 j 位的最长公共子序列。时间复杂度 $\Theta(nm)$，爆。
赛博做法：设 $f_{i,j}$ 表示 B 的前 i 位中最长公共子序列长度为 j 最少需要 A 前 $f_{i,j}$ 位。预处理出 A 每个位置的下一个 $a, b, c, \cdots, z$ 的位置。转移可做到 $\Theta(1)$。时间复杂度 $\Theta(m^2 + 26n)$。
可见状态的设计对时间复杂度的影响巨大。

End

Thanks for your listening.