

“音符大师”解题报告

中山纪念中学 邓奕鹏

题目大意

你在打音游，这一关有 n 个音符，会从屏幕的最上方落下，你需要在屏幕最下方的线上打到这些音符。

判定线可以看作数轴的**正半轴**， n 个音符依次落下，第 i 个音符会落到 p_i 的位置。你的**两只手**可以分别覆盖两个长度为 L 的区间，准确来说，一只手可以覆盖 $[x, x + L]$ ，使得当音符落在这个区间（包括端点）的时候可以准确打到。

游戏开始时你的两只手覆盖了 $[0, L]$ ，在接下来的 n 个音符落下的间隙中，你可以将你的手任意移动，移动的时间忽略不计，从 $[x, x + L]$ 移动到 $[y, y + L]$ 需要耗费 $|x - y|$ 的体力。

游戏开始前，你想要知道最少耗费多少体力可以依次打到所有的音符。

数据范围

对于100%满足 $1 \leq n \leq 50000, 0 \leq L \leq 50, 0 \leq p_i \leq 10^9$

subtask1(15%) : $n \leq 200, p_i \leq 200$

subtask2(15%) : $L = 0$

subtask3(30%) : $L \leq 5$

subtask4(40%) : 无特殊限制。

时间限制：3000ms

空间限制：512MB

解题过程

接下来我们逐个子任务考虑。

subtask1(15%)

由于 n 和 p_i 都比较小，可以直接DP，设 $f[i][j][k]$ 表示当前到音符 i ，两只手的位置分别在 j, k 的答案。

时空复杂度 $O(np_{max}^2)$

subtask2(15%)

覆盖区间为一个点时，考虑一个简单的DP，设 $f[x][i]$ 表示一只手到了音符 x ，另一只手到了音符 i ，当前音符为 x 的答案。

有转移 $f[x][i] \rightarrow f[x+1][x], f[x+1][i]$ 。考虑用数据结构维护这个转移，需要维护前缀 min ，后缀 min ，支持单点修改和全局加，用线段树就可以简单实现。

时间复杂度 $O(n \log n)$

subtask3(20%)

此时覆盖区间比较短，考虑将subtask2的做法稍微扩展一下，设 $f[x][i][j][k]$ 表示当前到音符 x ，一个区间在 $[p_x + j - L, p_x + j]$ ，另一个区间在 $[p_i + k - L, p_i + k]$ 的位置。转移与上面类似，在此就不赘述了。

时间复杂度 $O(nL^2 \log n)$

subtask4(40%)

我们可以考虑优化一下状态，如果只保留状态 $f[x][i][j][k], j \in [0, 1], k \in [0, 1]$ 表示当前到音符 x ，两个区间在 $[p_x + (j - 1)L, p_x + jL], [p_i + (k - 1)L, p_i + kL]$ 时的答案，即只保留区间端点在某个 p_i 时的状态，该怎么转移？

容易想到的是考虑第一个不在上面两个区间的音符 $y(y > x)$ ，分别转移到 $f[y][x]$ 和 $f[y][i]$ 即可。

同样考虑用数据结构优化它，考虑将 i, k 两个状态放到线段树上，那么考虑 $f[x][j = 0/1]$ 的区间 $[p_x + (j - 1)L, p_x + jL]$ ，后面第一个不在该区间内的音符为 y ，那么可以将线段树中代表的区间分为**包括 y 的**和**不包括 y 的**两个部分。

转移1：不包括 y 的区间可以直接在线段树上转移（类似subtask2）。

转移2：包括 y 的最多只有 $L + 1$ 个，我们可以将这些区间提取出来，即暴力做 $[p_x + (j - 1)L, p_x + jL], [p_i + (k - 1)L, p_i + kL]$ 区间对的转移，即找到不在内部的第一个 y ，转移过去。

可以用动态开点线段树实现，需要支持合并。

时间复杂度转移1 $O(n \log n)$ ，转移2 $O(nL \log n)$ 。

空间复杂度最大为 $O(nL \log n)$ ，可以通过回收减小空间，比较难将其卡满。

出题人不排除会有其他的做法，如果有的话欢迎讨论。

参考资料

暂无