IOI2022 集训队试题准备

冯施源

1 题目大意

有 n 个事件，每个事件 (t_i, x_i) 表示在 t_i 时刻，x_i 处会降落一颗球。

小 F $t_0 = 0$ 时刻在 $x_0 = 0$，现在要去接这些球，要求要么小 F t_i 时刻在 x_i，要么小 F 的分身在。若当前时刻小 F 在 x，那么下一时刻他可以移动到 $x-1, x$ 或 $x+1$。小 F 可以在任意时刻，在他所处位置放下一个不能移动的分身，可以用分身来接球，但当放下一个分身时，之前存在的分身会在 0.01 时刻后消失。

问小 F 能不能接到所有球。

2 数据范围

128MB，1s。

本题有 8 个子任务

Subtask 1（5%），$n \leq 20$，特殊性质。

Subtask 2（5%），$n \leq 100$，依赖子任务 1，特殊性质。

Subtask 3（10%），$n \leq 5000$，依赖子任务 2，特殊性质。

Subtask 4（20%），$n \leq 5000$。

Subtask 5（20%），$n \leq 10^5$，依赖子任务 3，特殊性质。

Subtask 6（10%），$n \leq 3 \times 10^5$，依赖子任务 5，特殊性质。

Subtask 7（20%），$n \leq 3 \times 10^5$，依赖子任务 4,6。

Subtask 8（10%），$n \leq 10^6$，依赖子任务 7。

对于全部数据，满足 $n \leq 10^6$，$\forall i \geq 1, t_i \geq t_{i-1}, |x_i|, |t_i| \leq 10^9$。

特殊性质: 满足 x_i 互不相同。

3 解题过程

这道题是笔者加强的 CF1456D，官方题解给出的做法是算法 2。

3.1 算法 1

直接枚举哪些点是用分身接住的，我们注意到分身放置的顺序一定是按事件的时间递增的，这时我们一个点一个点跑，看当前点接住后能不能抽空去下一个分身点放置分身。

复杂度 $O(n2^n)$，期望得分 10 分。

3.2 算法 2

这个算法是官方题解的做法，可以 $O(n^2)$ 解决 x_i 互不相同的情况。

我们考虑设计 DP，设 f_i 表示之前的都接完了（包括分身的点），到 i 的最小时间。

那么我们知道此时的最优策略就是在 i 放一个分身，然后去完成其他任务。若直接去接 $i+1$，我们可以用 $\max(f_i+|x_{i+1}-x_i|,t_i)$ 来更新 f_{i+1}。还有情况就是从 i 跑到 x_j，放下分身，再到 $i+1$。此时我们注意到在 (i,j) 之间一定都不是用分身接的，由此新增状态 $g_{p,q}$ 表示人在 x_p，分身在 x_q 是否合法。那么上述情况，若 $\max(f_i+|x_j-x_i|,t_i)+|x_j-x_{i+1}|\le t_{i+1}$，我们可以将 $g_{i+1,j}$ 设为合法。

下面讨论 $g_{p,q}$ 的转移，若 $p+1<q$，那么转移到 $g_{p+1,q}$ 就可以了（因为 $p+1$ 一定用本体接住）。若 $p+1=q$，那么有两种，一种是从 p 走到 j，放分身，再到 $q+1$，可以根据合法性更新 $g_{q+1,j}$，或者直接走到 $q+1$，更新 f_{q+1}。

复杂度 $O(n^2)$，可以获得 20 分。

3.3 算法 3

算法 3 讨论的也是 $O(n^2)$ 解决 x_i 互不相同的情况。

算法 2 的状态数就为 $O(n^2)$，看起来不太方便优化。我们考虑延续算法 1。我们枚举哪些点是用分身接的，设 f_i 表示在保证前 $i-1$ 个点都顺利拿下的情况下，第 i 个点放下分身的最早时间是多少。根据后续的实现细节，需要知道一个分身点前面是否为分身点，于是设 $f_{i,0/1}$ 表示 $i-1$ 是否为分身点。

考虑转移，我们枚举前一个分身点的位置，若为 $i-1$，讨论如下:

1. 从 $f_{i-1,0}$ 转移，此时我们知道 t_{i-2} 时一定在 x_{i-2}，且 $f_{i-1,0}$ 时刻在 x_{i-1} 上，那么我们只需要用 $\max(t_{i-2}+|x_i-x_{i-2}|,f_{i-1,0}+|x_i-x_{i-1}|,t_{i-1})$ 来转移到 $f_{i,1}$。

2. 从 $f_{i-1,1}$ 转移，同上，此时我们知道此时只需要用 $\max(f_{i-1,1}+|x_i-x_{i-1}|,t_{i-1})$ 来更新 $f_{i,1}$ 即可。

接下来, 考虑不从 $i-1$ 转移的情况, 枚举上一个分身点 $j \le i-2$, 我们要求从 $j+1 \to i-1$ 都用本体接，那么就要求 $\forall k \in [j+1, i-2]$ 满足 $t_{k+1} \ge t_k + |x_{k+1} - x_k|$。

若从 $f_{j,0}$ 转移，此时要先到 x_i 放分身，再走到 x_{j+1}，在 i 放分身的最早时间是 $T_j = \max(t_j, f_{j,0} + |x_i - x_j|, t_{j-1} + |x_i - x_{j-1}|)$，那么还要求 $T_j + |x_i - x_{j+1}| \le t_{j+1}$，若满足，则可以更新 $f_{i,0} = T_j$。若从 $f_{j,1}$ 转移，此时最早时间为 $T_j = \max(t_j, f_{j,1} + |x_i - x_j|)$，同理。

从 $f_{j,0}$ 转移的正确性容易验证，但为什么从 $f_{j,1}$ 转移是正确的呢?

因为有 $x_j \ne x_{j-1}$，故蕴含着 $f_{j,1}$ 时刻在 x_j 放下分身这一信息，并且 $f_{j,1} \ge t_{j-1}$，所以是可以直接用 $f_{j,1} + |x_i - x_j|$ 的。期望得分 20 分。

3.4 算法 4

这个做法是算法 3 的扩展，可以 $O(n^2)$ 解决有 x 相同的情况。

我们考虑扩展 $f_{i,0/1}$ 的定义，表示为 i 为分身点，记 pre_i 表示 $\max\{k \mid x_k \ne x_i, k < i\}$，0/1 表示 pre_i 是否为分身点。

考虑转移，先考虑从 $j \le i-2$ 转移的情况，且 $x_j \ne x_i$。

1. 从 $f_{i,0}$ 转移，此时在 x_i 放下分身的最早时间

$T_j = \max(t_j, f_{j,0} + |x_i - x_j|, t_{pre_j} + |x_i - x_{pre_j}|)$。

2. 从 $f_{i,1}$ 转移，此时在 x_i 放下分身的最早时间 $T_j = \max(t_j, f_{j,1} + |x_i - x_j|)$

这里更改定义后，有 $f_{j,1} \ge t_{pre_j}$ 了，拥有了同算法 3 的性质。

下面来讨论 $x_j = x_i$ 的情况，那么我们只需要保证能在 t_{j+1} 前走到 x_{j+1}，我们就可以用 f_j 来更新 f_i。讨论如下:

1. 若从 $f_{j,0}$ 转移，则要求 $T_j = \max(f_{j,0} + |x_{j+1} - x_j|, t_{pre_j} + |x_{j+1} - x_{pre_j}|) \le t_{j+1}$

2. 若从 $f_{j,1}$ 转移，则要求 $f_{j,1} + |x_{j+1} - x_j| \le t_{j+1}$

最后来讨论来自 $i-1$ 的转移。

1. 当 $x_i = x_{i-1}$ 时，我们只需要执行 $f_{i,0/1} = f_{i-1,0/1}$ 即可。

2. 当 $x_i \ne x_{i-1}$ ，从 $f_{i-1,0}$ 转移，即将 $\max(t_{pre_{i-1}} + |x_i - x_{pre_{i-1}}|, f_{i-1,0} + |x_i - x_{i-1}|, t_{i-1})$ 转移到 $f_{i,1}$。

3. 当 $x_i \ne x_{i-1}$，从 $f_{i-1,1}$ 转移，即将 $\max(f_{i-1,1} + |x_i - x_{i-1}|, t_{i-1})$ 转移到 $f_{i,1}$。

复杂度 $O(n^2)$，期望得分 40 分。

3.5 算法 5

观察算法 4 的转移，设 $T_j = \max(t_j, f_{j,0} + |x_i - x_j|, t_{pre_j} + |x_i - x_{pre_j}|)$，并且要求 $T_j + |x_i - x_{j+1}| \le t_{j+1}$，这告诉我们，每个 j 都限制了 x_i 在一个区间，在区间内才能转移。并且转移的代价是关于 x_i 的分段函数，每段的代价可能是如下的几种：$-x + c, c, x + c$，对这 3 种分别维护 c 的最值，那么对每个 j，我们在对应可以转移的区间进行修改，转移时单点查询 x_i 上的 max，用线段树就可以解决。

对于从 $x_j = x_i$ 转移的情况，只需要对 $T_j = \max(f_{j,0} + |x_{j+1} - x_j|, t_{pre_j} + |x_{j+1} - x_{pre_j}|) \le t_{j+1}$ 或 $f_{j,1} + |x_{j+1} - x_j| \le t_{j+1}$ 的 j 记录下 $f_{j,0/1}$ 的最小值即可。

复杂度 $O(n \log n)$，但是分段的细节较多，常数较大，期望得分 $90 \sim 100$ 分。

3.6 算法 6

我们来考察 T_j 和 T_k 的关系($j < k$)。我们知道,从 $f_{j,0}$ 转移的话,需要 $T_j + |x_i - x_{j+1}| \le t_{j+1}$，但由于 $T_k \ge t_k \ge t_{j+1}$，我们知道，$T_j \le T_k$，这样的话，我们只需要找到能转移的第一个 j 进行转移就可以了。问题变成了每次丢进去一个区间，对没覆盖的部分覆盖，用并查集就可以解决。不需要进行函数分段的讨论。

复杂度除离散化之外 $O(n\alpha(n))$，期望得分 100 分。

4 参考资料

官方题解：https://codeforces.com/blog/entry/85118