傅里叶与交通规划 解题报告

题目大意

平面上给定 n 个传送带，每个传送带占据了 x 坐标处于 $[p_{i-1},p_i)$ 内、y 坐标为全体实数的区域。人处在传送带上时，y 坐标会以 v_i 单位每秒的速率变化，v_i 可正可负。除了传送带以外，人还可以以不超过 V 的速率，向平行于任一坐标轴、可以为正亦或为负的方向移动，与传送带的移动叠加。人的运动状态可以在任何时刻立刻切换。可以走到传送带外，此时没有传送带的强制位移。

给定 q 次询问，每次询问从一个坐标 (x_1,y_1) 到达另一个坐标 (x_2,y_2) 所需最小时间。相对误差或绝对误差不超过 10^{-5}。

数据范围

对于所有数据，均满足 $n,q\le 1.5\times 10^5$，$-5\times 10^5\le p_0<p_1<p_2<\cdots<p_n\le 5\times 10^5$，$|x_1|,|y_1|,|x_2|,|y_2|\le 5\times 10^5$，$0\le |v_i|<V\le 5\times 10^5$。所有输入均为整数。

Subtask 1(5%): 保证 $n=0$。

Subtask 2(10%): 保证 $n,q\le 1000$。

Subtask 3(10%): 保证对于所有询问，$x_1=p_0,x_2=p_n$。

Subtask 4(10%): 保证所有询问的 x_1 全都相等，所有询问的 x_2 全都相等（但不保证 $x_1=x_2$）。

Subtask 5(15%): 保证 v_i 单调不降，且询问的 $x_1\le x_2$。

Subtask 6(15%): 保证存在 i 使得 $p_i=x_1=x_2$。（但并不保证所有询问的 i 都相同）

Subtask 7(15%): 保证除 n,q,V 外，其它值都在合法范围内独立随机得到（p 的随机方式是随机 $n+1$ 个不等的 $[-5\times 10^5,5\times 10^5]$ 中的值并排序）。

Subtask 8(10%): 保证 $n,q\le 5\times 10^4$。

Subtask 9(10%): 无特殊限制。

解题报告

出实数题真是抱歉。在这里向所有被卡精度的人（如果存在的话）表示诚挚的歉意。当然，std 是全程使用整数、分数的，不存在误差，只不过为了方便大家写代码所以标准答案是实数。

- subtask 1 是作为基础的暴力分存在的，直接计算曼哈顿距离并除以 V 即可。

以下的 subtask 就需要基本的分析了。

首先必然要从起点的 x 坐标走到终点的 x 坐标，这段路程是不可省、且仅可能通过步行来在 x 轴上移动的，将这段路程上传送带对 y 轴的影响计入位移。剩下的位移必然贪心地选取路程中最优的那个传送带完成（如果还需要正的位移，就找到 v 最大的那个，以 $v+V$ 的速度向 $y+$ 移动；否则需要负的位移，找到 v 最小的那个以 $v-V$ 的速度向 $y-$ 移动）；但不管怎么说，最优方案里，我们定然只会在一个传送带上调整 y 坐标。于是，我们可以钦定一个传送带为最优传送带，使用最短路径走到其然后再到终点，则最优的方案必然可以写成上述方案之一。

> 更加严谨的证明：

> 首先，第一个观察是我们其实对 y 轴坐标并没有那么在意。因此我们将主要的精力放在 x 轴上。对于任意一条路径，我们可以将其中选择向 x 轴方向步行的时刻和向 y 轴方向步行的时刻分拆出来：所有选择沿 y 轴方向步行的时刻，x 轴上都是没有位移的；这意味着我们可以首先删去那些 y 轴方向步行的时刻，得到一条每时每刻都仅在 x 轴上步行的路径，这条路径离我们目标的路径仅差若干 y 轴上的位移。
>
> 假如 x 轴步行方案已经确定，我们可以使用 y 轴步行来调整 y 轴坐标。通过 x 轴步行方案，我们可以简单求出当前方案内 y 轴的位移（这些位移仅由传送带贡献，而其可以简单通过在每个传送带停留的时长求出），与我们目标的 y 轴位移对比后，得到差值即为我们需要用 y 轴步行调整的量。
>
> 假如差值为零，则显然当前仅在 x 轴步行的方案就是一组合法的方案。否则，若差值为正，我们贪心地找到当前路径中我们曾访问过的全体传送带中，v_i 最大的一个，并在上面使用 V 的最大速度向 y 轴正方向移动，以 $v_i + V$ 的最大合速度补齐差值。同理，若差值为负，我们即找到 v_i 最小的一个并在上面以最大速度向负方向移动。
>
> 我们发现一组 x 轴步行方案中，我们仅在意三个值：当前步行方案与目标位移的差值 D；当前步行方案中，根据 D 的符号决定的最大速度或最小速度 v；当前步行方案目前消耗的时长 T。
>
> 在 D 的符号以及 v 均确定的时候，我们可以尝试精简一下步行方案。首先，如果存在一组方案，使得在不改变 D 的符号和 v 的前提下，减小了在某个传送带上沿 x 轴步行的时间，则这组方案一定是更优的：因为你减小的时间原本会以不优的 v_i 速度被动地向目标的 D 移动，现在以最优的 $v + V$ 速度移动，定然更优。
>
> 进一步，就算你减少时间使得 D 变号了，仍然是更优的，因为你减小时间对于 D 的影响是连续的，有一时刻 $D = 0$，此时不论 v 取何值都无影响，所以可以在此时调整 v 为 D 变号后的 v'，进而继续调整，（相对于调整后的 $D = 0$ 和 v'）D 的符号和 v' 都是不变的。
>
> 按照上述分析，则考虑任一组最优的答案，就会发现其所有时间都被压榨到了极限，也即其整个流程即为起点到最优的 v 然后到终点。考虑在 v 处调整 y 轴位移的过程后，流程是：起点 $\to$ 最优点；最优点调整位移；最优点 $\to$ 终点。第一、第三部分仅包含 x 轴上最大速度前进，而第二部分仅包含 y 轴上最大速度调整。
>
> 于是，我们只需考虑全体满足上述流程的方案即可。

- subtask 2 可以直接枚举钦定的最优传送带是哪个。可能还存在一些其它的 n^2 级别的算法。

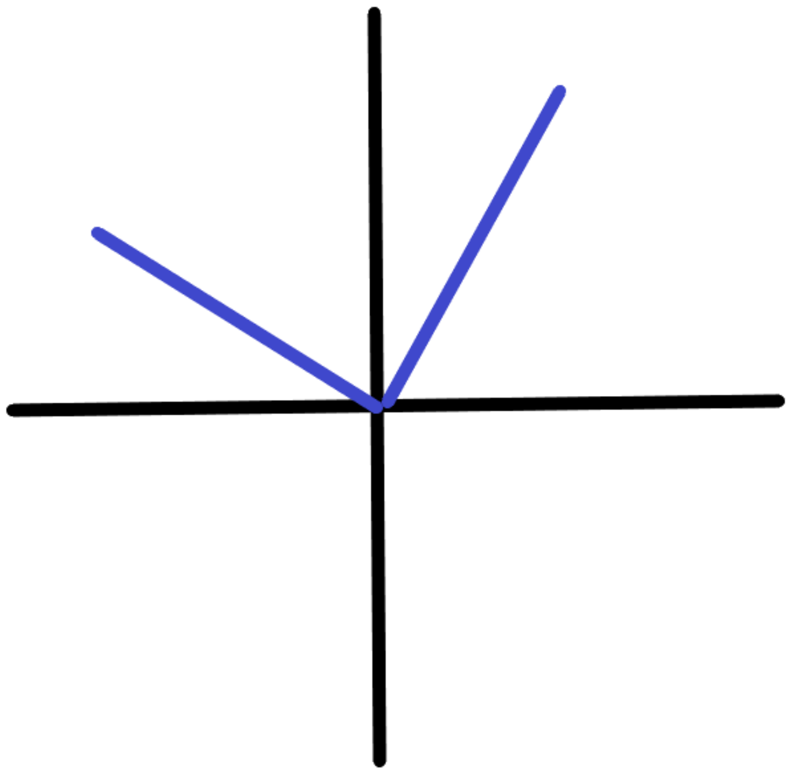
如果最优传送带介于起讫点 x 坐标之间，则最短路径即为起点到终点直接走的方案，计算得知此时还需要的位移后，找到该位移对应的最大 v 或最小 v 在其上运动即可。

- subtask 3 只需考虑此种情形，而 subtask 5 则不需考虑此种情形。

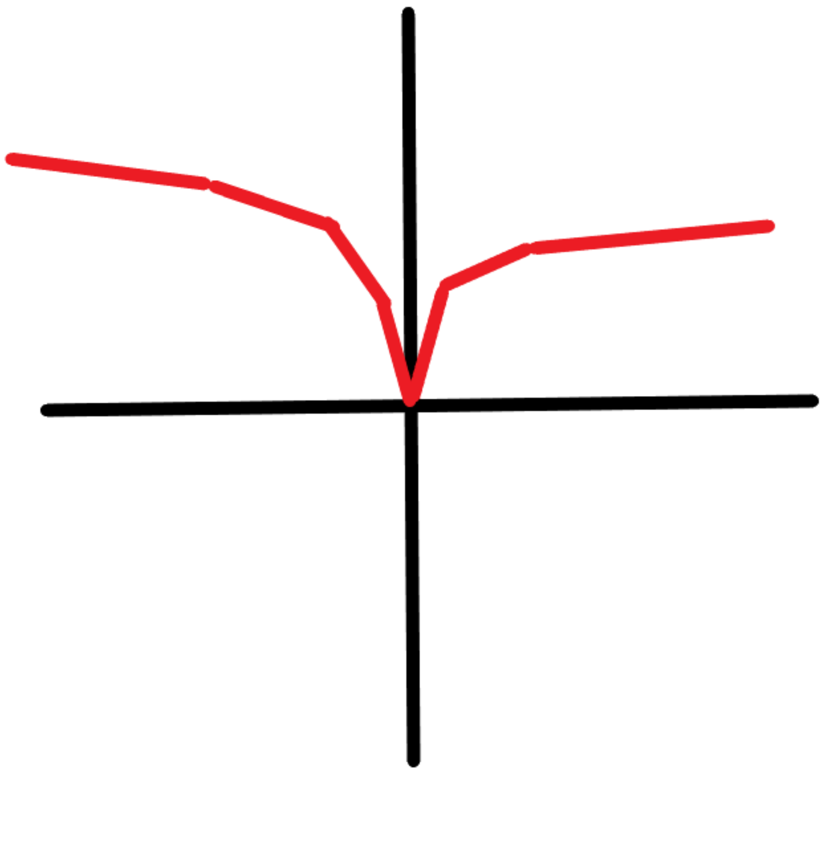
否则该最优传送带位于起讫点 x 坐标之外，不妨令起点在终点左侧，而最优传送带更在起点左侧，则该流程可以写作：自起点出发，一路向左走到该最优传送带后，在最优传送带上走若干时间直到补足位移，然后返回。在事先统计好起点到终点间的位移后，这可以被写成如下的二元组 (x, Δ)，其中 x 为起点左侧的首个传送带，意为自 x 向左走到某个最优传送带然后返回，总计贡献 Δ 的 y 轴位移，所需要的时间。

- subtask 7 未能想到上述问题正解的人也可以通过：基于跑得更远必然为了一个更优（指更大/更小）的传送带，所以要回答 (x, Δ) 的问题只需访问关于速度的后缀 $\min / \max$，也即单调栈上的元素，随机数据下期望各有 $\log$ 个元素。
- subtask 6 则就是上述问题。这一档部分分的意图是希望启发选手根据这个问题来逆推出开头关于路径特征的分析。

现在我们来解决上述问题。我们令 $f_x(\Delta)$ 表示该二元组对应的最小时间。若钦定最优传送带为 x 自身，那么 $f_x(\Delta)$ 在 $\Delta > 0$ 时即为 $\dfrac{\Delta}{v_x + V}$，$\Delta < 0$ 时即为 $\dfrac{\Delta}{v_x - V}$，构成如下的两根射线构成的图像（其中横轴为 Δ，纵轴为 $f(\Delta)$）：

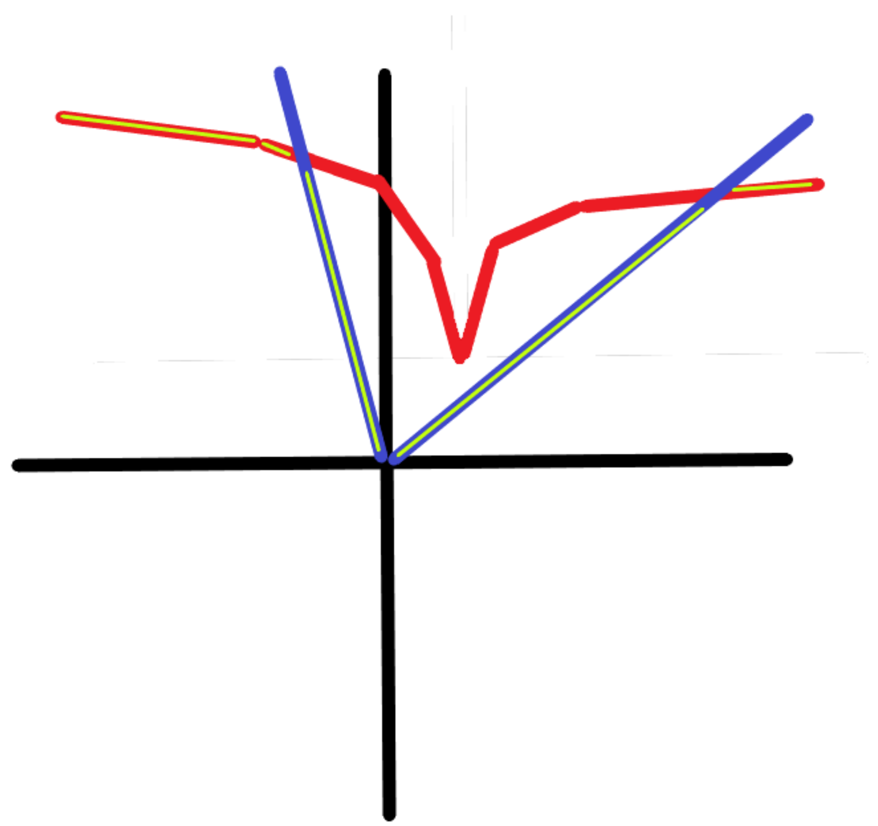


现在，我声称，若令最优传送带可以为 x 左侧的任一传送带，则 $f_x(\Delta)$ 分别构成 $\Delta > 0$ 和 $\Delta < 0$ 的两个凸壳，即下图的模式：



这可以被归纳证明：

- 首先 $f_0(\Delta)$ 只由两根射线组成，即为我们一开始提到的图像。显然这满足两个凸包的模式：单一的射线也构成凸包。
- 然后 $f_x(\Delta)$ 可以是自身的两条射线，也可以是 $f_{x-1}(\Delta)$ 经过转移得到。
- $f_{x-1}(\Delta)$ 的转移，相当于是将整个图像向上、向左/右平移了一段距离。
- 于是 $f_x(\Delta)$ 即为自身的两条射线与平移后的 $f_{x-1}(\Delta)$ 两个函数的 $\min$。
- 平移的 Δ 坐标变化量为 $\dfrac{2(p_i-p_{i-1})v_i}{V}$，$f(\Delta)$ 坐标变化量为 $\dfrac{2(p_i-p_{i-1})}{V}$，其与原点连线斜率为 $\dfrac{1}{v_i}$。我们发现，平移后的 $f_{x-1}(\Delta)$ 有着好的性质：其原本的原点平移后的位置，必然被"夹在" $f_x(\Delta)$ 自身两条射线的中间，即下图的模式：



- 上图中绿色的部分即为二者的 $\min$，即 $f_x(\Delta)$。
- 可以发现，这相当于平移后，对两侧凸壳分别各自插入了一条新的直线构成的新结构，在弹出不再合法的直线后其仍是凸壳。并且，因为插入的新直线过原点，所以两侧的凸壳仍然过原点。

于是在 $\Delta+$ 和 $\Delta-$ 分别维护一个凸壳，平移可以通过打 `tag` 解决，插入就是经典的不断 `pop` 直到成凸壳。询问 $f_x(\Delta)$ 的时候直接在对应侧的凸壳上二分即可。

- subtask 4 作为 subtask 3 的延伸，此时只需对起讫点两侧的 x 各自维护完整的 $f_x(\Delta)$，可以用一些例如扫描线之类的数据结构维护这个图像。同时也启发大家把图像画出来，以得到其更多信息。

- 总复杂度 $O(n+q\log n)$，假如对最优传送带夹在起讫点之间的场合使用一些高级的 RMQ 数据结构以求出最大/最小的 v 的话。std 使用了简单的 st 表，因此复杂度是 $O((n+q)\log n)$ 的。这是理论的、可以通过全部 subtask 的正解。
- 本来计划的数据范围是 $n,q\le 5\times 10^5$，但是考虑到数据包大小的限制所以最终将数据范围定为 1.5×10^5。肖子尧同学提供了一种 $n\log^2 n$ 的李超树做法，考虑到给予类似复杂度的算法合理的分数，所以增加了 subtask 8 的档次。事实上，实现良好的 $n\log^2 n$ 做法也不乏通过所有分数的可能性。

上述做法可以扩张到 $v_i \geq V$ 的场合，此时一侧凸包可能不存在。但是为了减小代码量，我们保证了 $v_i < V$，此时两侧凸包必然均存在。

上述做法可以使用实数进行，但是为了规避精度问题，我们可以尝试避免使用实数。

首先，如果钦定的最优传送带在起讫点之间，那么此时的 Δ 和 T 均可以被写成 $\dfrac{z}{V}$ 的形式，其中 z_Δ 是 2.5×10^{11} 级别的整数，z_T 则是 5×10^5 级别的整数。$\dfrac{\Delta}{v+V}$ 则是一个分子、分母均是 2.5×10^{11} 级别的整数，$\dfrac{\Delta}{v+V} + T$ 亦然。比较分数大小时可能需要 `__int128`。

然后是不在起讫点间的场合。此时的 (x, Δ, T) 三元组中，Δ, T 的级别和前述一致。求凸包时，每次水平平移和竖直平移的量都是 $\dfrac{z}{V}$，直线的斜率是 $\dfrac{1}{z}$。因为在凸包中询问的 Δ 也是 $\dfrac{z}{V}$ 的形式，所以可以将 Δ 坐标、$f(\Delta)$ 坐标统一扩大 V 倍，这样其中的一条射线就可以被写成 $\dfrac{\Delta + b}{k}$ 的形式，其中 k, b 都是整数。这样，维护凸包的过程也可以规避实数，那么整份代码就可以取缔实数，实现求得精确解的效果。

参考文献

[OI-wiki：凸包]