动态规划的斜率优化

于志竟成

2015 年 4 月 11 日

加减：$(x_1, y_1) \pm (x_2, y_2) = (x_1 \pm x_2, y_1 \pm y_2)$

取模：$|(x, y)| = \sqrt{x^2 + y^2}$

点积

$$(x_1, y_1) \cdot (x_2, y_2) = x_1 x_2 + y_1 y_2$$

$$\vec{a} \cdot \vec{b} = |\vec{a}| \left|\vec{b}\right| \cos\theta$$

$$\vec{a} \perp \vec{b} \iff \vec{a} \cdot \vec{b} = 0$$

叉积

$$(x_1, y_1) \times (x_2, y_2) = x_1 y_2 - x_2 y_1$$

$$\vec{a} \times \vec{b} = |\vec{a}| \left|\vec{b}\right| \sin\theta$$

。这里的 θ 是有方向的，指的是从 $\vec{a}$ 转到 $\vec{b}$ 的角。

$$\vec{a} \parallel \vec{b} \iff \vec{a} \times \vec{b} = 0$$

。

叉积可以用于判断一个向量相对于另一个向量的方向。

凸包

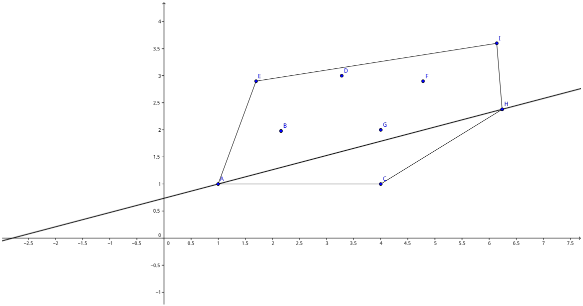
一个点集是凸集当且仅当其中任意一对点的连线上的点都在这个点集中。

给定 N 个点，这些点的凸包就是包含这 N 个点的最小凸集。

二维凸包可以在 $O(N \lg N)$ 时间内求出。

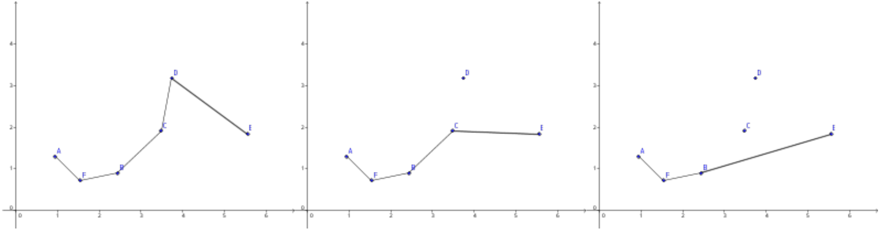
显然，凸包的顶点一定在这 N 个点中。

可以将凸包看作上下两部分。两部分拼在一起就是整个凸包。



最左边的点显然在凸包上（如果有横坐标相等的取最高和最低的点）。考虑分别求出上凸包和下凸包。将所有点按照横坐标排序后，按此顺序一个一个加入凸包（其实可以说是增量的思想，就是求出前 k 个点的凸包，现在求前 $k+1$ 个点的凸包）。上下凸包的相邻点连线的斜率都是分别单调的，由于新加入的点是当前所有点最靠右的，它也必然在当前的凸包内（先不考虑横坐标相等）。为了保证斜率单调，在发现前一个点不满足条件时需要将前一个删去。这里判断就需要用到叉积。

下凸包的情况：



用单调栈就行了。

稍微想一下就可以发现中间的点出现横坐标相等的情况是毫无影响的。

给定 N 个整数 x_i 和 a, b, c 三个整数，将 N 分成任意连续段，对于和为 s 的段，其贡献为 $as^2 + bs + c$。求最大贡献和。

$N \leqslant 1000000, -5 \leqslant a \leqslant -1, |b|, |c| \leqslant 10000000, 1 \leqslant x_i \leqslant 100$。

$O(N^2)$ 的动态规划

用 $f(i)$ 表示将前 i 个数分段的最大贡献和，$s(i)$ 表示前 i 个数的和。很容易得到方程

$$f(i) = \max_{j=0}^{i-1} \left(f(j) + a\left(s(i) - s(j)\right)^2 + b\left(s(i) - s(j)\right) + c \right)$$

。直接做是 $O(N^2)$ 的。

必须进行优化。

如何优化？

状态数是 $O(N)$ 的，无法优化。考虑优化转移。

需要对转移方程进行研究。

假设 $k < j < i$，考虑 $f(i)$。k 优于 j 当且仅当

$$
\begin{aligned}
f(k) + a(s(i) - s(k))^2 + b(s(i) - s(k)) + c > \\
f(j) + a(s(i) - s(j))^2 + b(s(i) - s(j)) + c
\end{aligned}
$$

。

假设 $k<j<i$，考虑 $f(i)$。k 优于 j 当且仅当

$$\begin{aligned} f(k)+a(s(i)-s(k))^2+b(s(i)-s(k))+c > \\ f(j)+a(s(i)-s(j))^2+b(s(i)-s(j))+c \end{aligned}$$

。变形可以得到

$$\frac{\left(f(j)+as(j)^2-bs(j)\right)-\left(f(k)+as(k)^2-bs(k)\right)}{2as(j)-2as(k)} > s(i)$$

。

令不等式左边为 $h(k,j)$，下面探究 h 的性质。

现在 k 优于 j 当且仅当

$$h(k,j) > s(i)$$

。

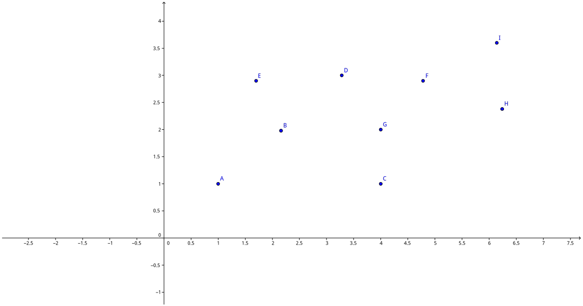
对于 $x < y < z < i$，如果 $h(x,y) > h(y,z)$，那么 y 要么不比 x 优，要么不比 z 优（注意到这与 i 无关）。

分两种情况考虑：$h(x,y) > s(i)$ 时，x 优于 y；$h(x,y) \leqslant s(i)$ 时，$h(y,z) < s(i)$，z 优于 y。

所以 y 可以直接扔掉不考虑，因为不论当前计算哪个 $f(i)$ 它都不可能用上。

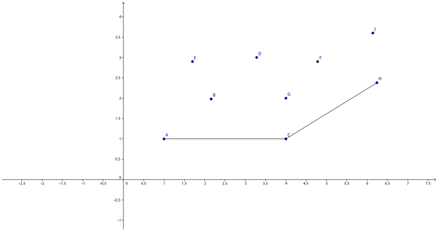
下面来点形象的解释。

定义点 p_i 为 $\left(2as(i), f(i)+as(i)^2-bs(i)\right)$，于是有了一堆点，比如说



注意 $p_k, p_j (k < j < i)$ 连线的斜率就是 $h(k, j)$。最优的点是哪个呢？

其实可以当作有一条斜率为 $s(i)$ 的直线卡在某个点（所有其他点都在直线上或其上方），这个点就是最优的。



注意到那些凹进去点永远也不可能“卡住”一条直线，所以直接删掉。最后我们会发现我们只需要关心下凸包上的点。于是维护之。

怎么维护呢？

注意到那些凹进去点永远也不可能“卡住”一条直线，所以直接删掉。最后我们会发现我们只需要关心下凸包上的点。于是维护之。

怎么维护呢？由于点的横坐标 $2as(i)$ 单增，直接将其插入凸包的单调栈就行了。

如何回答询问（即给定 $s(i)$ 求最优点）？下凸包上相邻点的斜率单调递增，所以相当于要找 $s(i)$ 的临界位置。注意到本题中 $s(i)$ 单增，所以最优点标号不减，无用的点直接从头部删去就行了。

于是最后的做法是：用单调队列维护下凸壳，每次询问将头部一部分扔掉。复杂度 $O(N)$。

更一般的情况（不详述）

这道题用到了很多性质，例如横坐标单增、询问的斜率单增。

对于更一般的情况，如果点的横坐标不单调，就需要用数据结构维护凸包而不能仅仅用单调栈（队列）；如果询问的斜率不单调，就需要对每次询问用二分查找。

给一条 N 个点的链，每条给定一个正整数长度。每个点有两个参数 a_i, b_i，从点 i 出发向前到达距离为 d 的点的代价为 $a_i d + b_i$。对每个点求出从它出发到达第一个点的最小代价。

$N \leqslant 10^6$。