Yufeng's 最近点对算法

Shuxin Chen, 2015013229

2017 年 3 月 2 日

1. 什么是 Yufeng's 最近点对算法？它的时间复杂度是多少？

   假设我们在实数平面 $S = \{(x, y)| - C \leq x \leq C, -C \leq y \leq C\}$ 上独立随机生成 n 个点，我们可以发现，平面 S 是一个边长为 $a_S = 2C$ 的正方形。

   我们现在可以把这个大正方形平面划分成若干个小正方形。令 $a_T = \frac{a_S}{\lfloor\sqrt{n-1}\rfloor} = \frac{2C}{\lfloor\sqrt{n-1}\rfloor}$，我们把平面 S 的一边划分成 $\lfloor\sqrt{n-1}\rfloor$ 条相等的线段。若将平面 S 的两条边都以此方式划分，我们可以将 S 划分成 $\lfloor\sqrt{n-1}\rfloor^2 \leq n-1$ 个边长为 a_T 的小正方形。

   根据鸽巢原理，当有 n 个点时，会有两个点落在同一个小正方形中，此时它们之间的距离最大为 $d_{max} = \sqrt{2}a_T = \frac{2\sqrt{2}C}{\lfloor\sqrt{n-1}\rfloor}$。因此，平面 S 上最近点对的距离 d_{ans} 总会满足 $d_{ans} \leq d_{max}$。

   我们可以根据这条性质来设计一种时间复杂度小于 $O(n^2)$ 的算法。首先我们将所有的点按照 x 轴排序，然后和朴素的算法类似，使用两重循环枚举两点 p_i, p_j，计算它们的距离并与当前的最短距离进行比较。在枚举时，我们固定 i，若当前枚举到的 p_j 与 p_i 的 x 坐标的距离已经大于 d_{max}，由于点已经排好序，因此 p_j 之后的点与 p_i 的 x 坐标的距离一定也大于 d_{max}，此时可以放弃 j 而开始枚举下一个 i。

   设每个点 p_i 需要枚举的 p_j 的期望个数为 $\mathbb{E}(i)$，由随机生成的独立性，我们有

$$\begin{aligned}
\mathbb{E}(i) &= \sum_{j=1}^{n} Pr(|x_j - x_i| \leq d_{max}) \\
&= n * Pr(|x - x_i| \leq d_{max}) && (x \in [-C, C]) \\
&= n * Pr(x_i - d_{max} \leq x \leq x_i + d_{max}) && (x \in [-C, C]) \\
&= n * \frac{2d_{max}}{2C} \\
&= n * \frac{2 * \frac{2\sqrt{2}C}{\lfloor\sqrt{n-1}\rfloor}}{2C}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{2\sqrt{2}n}{\lfloor\sqrt{n-1}\rfloor} \\
&= O(\sqrt{n})
\end{aligned}
$$

因此每个点期望枚举的点的个数是 $O(\sqrt{n})$ 的，因为排序的时间复杂度为 $O(n\log n)$，因此该算法的总期望时间复杂度为 $O(n\sqrt{n})$。