

NEERC2013 难题解题报告

长沙市雅礼中学 刘研绎

2013-2014 ACM-ICPC Northeastern European Regional Contest中的难题解题报告。

1 D. Dictionary

1.1 试题大意

为了给一些单词建立一个字典，我们需要建立一棵有根树，不妨称其为字典，每一条边用恰好一个字母标记，并且需要满足每一个单词都出现在树的路径上，即一条从某个节点往下到其某个子节点的路径上的字母串起来等于这个单词。现在给出 n 个单词，问给这些单词建立一个这样的有根树最少需要的节点数量，并且输出方案。

$1 \leq n \leq 50$ 。单词大小不超过10个字符。

1.2 算法介绍

考虑如果一个单词是另一个单词的子串，那么我们可以简单的删掉这个单词，因为只有另一个单词出现他就一定出现。

观察题目我们可以发现，单词需要出现在某个点到根的路径上的一部分。不妨假设前 $n-1$ 个单词已经形成了一个字典，当要加入第 n 个单词 c 时，我们需要找 c 的一个最长的前缀使得其出现在当前的字典中。考虑这个前缀由哪些单词组成，因为去除了包含因此最多由两个单词，即 a 的某个子串和 b 的前缀拼起来。这时候不妨调换一下 b 和 c 的加入顺序，这样就能保证每次加入的单词与当前存在的字典的公共部分是某一个单词的子串。

有了这个性质我们就不需要维护整个字典的形状了，我们只需要对于每一个单词找到一个前驱单词就行了。不难发现这其实是一个有向图的最小生成树，问题便迎刃而解。

值得注意的是本题需要输出答案，比较麻烦，不过数据很小，很多地方可以暴力。

时间复杂度： $O(n^3)$ 。

2 E. Easy Geometry

2.1 试题大意

给出一个 n 个顶点凸多边形，你需要找出一个在多边形内部的边平行于坐标轴的面积最大的矩形。输出其左下角与右上角的坐标。

$$1 \leq n \leq 10^5$$

2.2 算法介绍

一个面积最大的矩形肯定有一个点会卡在凸多边形的边界上，这样我们枚举这一个点，计算 x 和 y 轴方向最大的延伸距离再算出面积，取一个最大的就行了。

直接这样做根本算不出来，不妨考虑交换枚举顺序，我们先枚举矩形 x 轴方向的长度，再枚举矩形左下角的坐标。令 $(x, y_a(x))$ 为横坐标等于 x 时凸多边形上凸壳的点， $(x, y_b(x))$ 为下凸壳上的点，当我们知道 x 时可以由二分快速算出答案来。

令 $s_l(x)$ 为固定矩形 x 轴方向长度为 l ，左下角坐标为 x 的面积大小，那么：

$$s_l(x) = l \times (\min(y_a(x), y_a(x+l)) - \max(y_b(x), y_b(x+l)))$$

令 $s(l) = \max_x s_l(x)$ ，很显然 $s_l(x)$ 是凸函数所以可以三分求出最值。

又 $Answer = \max_l s(l)$ ，可以发现 $s(l)$ 仍然是凸函数。

不妨令 $a(l) = \max_x \min(y_a(x), y_a(x+l)) - \max(y_b(x), y_b(x+l))$ ，那么 $s(l) = l \times a(l)$ 。很显然 $a(l)$ 是减函数，并且是斜率递减的一次分段函数，所以 $s(l)$ 自然是凸函数了。所以我们可以三分直接算出答案。问题便迎刃而解。

时间复杂度： $O(n + \log^3 n)$

3 I. Interactive Interception

3.1 试题大意

这是一道交互式试题。

在数轴上有一个点，初始位置在 $[0, p]$ 之间的某一个整点，有一个在 $[0, q]$ 之间的速度 v 。每一秒你可以进行一次询问形如“check l r”并且得到一个回答即这个点在 $[l, r]$ 之间，或者不在。询问之后这个点的坐标就会增加 v 。

你的目标是在某一秒是得知点的确切位置，当你知道此时点的位置时，你就不需要询问了，将此时点的坐标输出即可。

你最多询问100次。 $1 \leq p, q \leq 10^5$

3.2 算法介绍

这个题并不是很难，可以说因为是交互的原因吧。

注意到 $p, q \leq 10^5$ ，最多询问100次，也就是说对于每次询问之后我们可以暴力扫一遍所有初始位置进行一些计算什么的，这就大大降低了这题的难度。

考虑对每个初始位置维护一个可能的速度区间，这样每次询问之后可以对每个初始位置暴力更新其速度区间，并且对于这个初始位置，当前可能的点的位置组成一个等差序列。对于每次询问，令所有初始位置的所有当前可能的点的位置中位数为 R ，我们询问当前点是否在 $[0, R]$ 之内，这样每次都会减少一半的可能状态，问题便迎刃而解了。

时间复杂度： $O(p \log(pq) \log(n))$