

Solution

Dez

2017 年 10 月 12 日

1 Star Way To Heaven

题目要求最大化最小距离。考虑二分答案判断是否合法。把每个星星看作一个圆，我们首先观察：只有一串圆把上下边界连接起来，我们才不能从起点到终点。而此时答案就是这一串圆中最大的圆的半径（画个图有助于理解）。这题有一个很神奇的解法，可以把它看作一个最小生成树的题去做。假设每个星星之间都有边，不必真的把边都连出来，使用 *Prim* 算法即可。又因为这里不能用堆去优化，我们每次找最小边都很不方便。有一种方法是记录每个点与已经在集合中的点的最小距离，每在点集中添加一个点，就用它的距离试着更新剩下的点，这样每次更新都是 $O(n)$ 的复杂度。每次选一个点也是 $O(n)$ ，直到上边界和下边界联通为止（把上边界和下边界看作两个点，距离特殊计算，联通用并查集判断）。最终复杂度 $O(n^2)$ 。

2 God Knows

极长上升子序列（占坑）。

3 Lost My Music

每个点都带一个权值 c ，题目要求出对于每个点 u ，找出一个祖先 v 最小化 $\frac{c_v - c_u}{dis(u, v)}$ 。首先给答案式变形，记 d 为节点的深度：

$$\frac{c_v - c_u}{dis(u, v)} = -\frac{c_u - c_v}{d_u - d_v}$$

我们把 (c_i, d_i) 看作一个坐标，那么答案式很像一个斜率的形式。

然后跳过一段（我不会证）。。。

然后就成为了求凸包切线，用单调栈求解。但这不是一个序列，而是在树上，所以我们的栈需要可持久化。又是一个很妙的办法，我们把栈弄成一个类似于链表的形式。记数组 $l[]$ 为当前下标在栈中下一个元素，再加上一个神奇的倍增跳栈，每次用倍增找出当前节点 u 前面最优的祖先 v ，用 v 更新 u 的答案，然后令 $l[u] \leftarrow v$ 。这样的正确性与具体转移需要前面的证明（意思就是我也不懂），我们就可以在 $O(n \log_2 n)$ 的复杂度内求得正解。