

Task

MiracleEEEE

December 5, 2017

Contents

1	知识点补全计划	1
1.1	省选 [0/7]	1
1.1.1	TODO 字符串 [4/6]	1
1.1.2	TODO 图论 [3/17]	1
1.1.3	TODO 数学 [4/25]	2
1.1.4	TODO 动态规划 [2/6]	3
1.1.5	TODO 计算几何 [2/9]	3
1.1.6	TODO 搜索 [0/3]	3
1.1.7	TODO 数据结构 [0/4]	4
2	做题计划	4
2.1	题目泛做	4
2.1.1	NOIP 题目泛做 [1/1]	4
2.2	杂	5
2.2.1	2017 年 11 月 [2/2]	5

1 知识点补全计划

1.1 省选 [0/7]

1.1.1 **TODO** 字符串 [4/6]

- ☒ 后缀数组
- ☐ 后缀自动机
- ☐ 后缀平衡树
- ☒ AC 自动机
- ☒ KMP 及扩展 KMP
- ☒ manacher

1.1.2 TODO 图论 [3/17]

- ☒ 双连通分量
- ☐ 最大流
- ☐ 费用流
- ☐ 最小割
- ☐ 带上下界的网络流
- ☒ 树剖
- ☐ LCT
- ☒ 点分治
- ☐ 边分治
- ☐ 动态树分治
- ☐ 树分块
- ☐ 虚树
- ☐ 仙人掌
- ☐ 朱刘算法
- ☐ 弦图
- ☐ 区间图
- ☐ 对偶图

1.1.3 TODO 数学 [4/25]

- ☒ 中国剩余定理
- ☐ 博弈论
- ☐ 拉格朗日乘子法
- ☐ 单纯型
- ☐ 辛普森积分
- ☒ 容斥原理
- ☐ 莫比乌斯反演
- ☐ BSGS
- ☐ 置换群
- ☐ FFT

- ☐ NTT
- ☐ 多项式求逆
- ☐ 二次剩余
- ☐ 多项式科技
- ☐ 积分
- ☐ 极限
- ☐ 微分
- ☐ 导数
- ☐ Ploya 定理
- ☐ 贝叶斯定理
- ☐ 杜教筛
- ☐ Pollard-Rho 圆锥曲线分解法
- ☒ 线性基
- ☐ Miller-Rabin 素性探测
- ☒ 高斯消元

1.1.4 TODO 动态规划 [2/6]

- ☐ 斜率优化
- ☐ 插头 DP
- ☐ 四边形不等式
- ☐ 斯坦纳树
- ☒ 数位 DP
- ☒ 区间 DP

1.1.5 TODO 计算几何 [2/9]

- ☒ 基础内容
- ☐ 凸包
- ☐ 三角剖分
- ☐ 旋转卡壳
- ☐ 半平面交
- ☐ picks 定理

- ☒ 扫描线
- ☐ 动态凸包
- ☐ 三维计算几何

1.1.6 TODO 搜索 [0/3]

- ☐ 模拟退火
- ☐ 爬山算法
- ☐ 随机增量法

1.1.7 TODO 数据结构 [0/4]

1. TODO 离线算法 [1/5]

- ☐ 莫队
- ☐ 树上莫队
- ☐ 单调莫队
- ☐ CDQ 分治
- ☒ 整体二分

2. TODO 平衡树 [1/3]

- ☐ rope
- ☒ Treap
- ☐ 替罪羊树

3. TODO 其他 [1/6]

- ☒ 主席树
- ☐ 线段树
- ☐ 划分树
- ☐ KD-Tree
- ☐ 块状链表
- ☐ 二维线段树

4. TODO 可持久化数据结构 [0/5]

- ☐ 平衡树
- ☐ 数组
- ☐ Trie 树
- ☐ 块状链表
- ☐ 动态仙人掌

2 錄题计划

2.1 题目泛做

2.1.1 NOIP 题目泛做 [1/1]

1. **DONE NOIP 2016 愤怒的小鸟** <2017-11-24 Fri>

状态压缩: 动态规划

Description

给出 $n \leq 18$ 个敌人坐标，每次可以可以消灭一条过 $(0, 0)$ 抛物线上的敌人，求最小次数。

Solution

n 很小考虑状压。最朴素的动态规划，用 $f[s]$ 表示消灭 s 中的敌人的方案数。枚举下一次消灭哪两个敌人，计算抛物线转移，时间复杂度 $O(n^3 2^n)$ ，还可以继续优化。我们可以预处理抛物线能消灭哪些敌人，时间复杂度变为 $O(n^2 2^n)$ 。但还不够。考虑第一个敌人在这个状态转移的状态中一定被某一条抛物线消灭，这样我们只考虑过第一个敌人的抛物线，枚举其他敌人转移，一定不会丢失最优解。时间复杂度变为 $O(n 2^n)$ 。

2.2 杂

2.2.1 2017 年 11 月 [2/2]

1. **DONE POJ 3693 Maximum repetition substring** <2017-11-24 Fri>

后缀数组:ST 表

Description

给出一个字符串，求最大重复子串（重复次数最多，如果存在多个，求字典序最小的那一个）。

Solution

后缀数组的应用。直接下手不好解决，不妨枚举一下循环节的长度 $|L|$ 。我们发现，任何一个循环节为 $|L|$ 重复子串总会包含至少两个 $s[0], s[|L|], s[2|L|], \dots$ 字符。那么考虑枚举两个相邻的上述字符，可以通过后缀数组 + ST 表 $O(1)$ 求出 LCP 的长度，但是最长公共子串的开头并不一定是我们枚举的字符，所以还要求出最长向前能匹配多少。这可以通过倒过来做一次后缀数组得到。那么我们现在有了一个极长区间，可以求得这个区间的循环节个数 k ，也就可以求出一个区间 $[l, r]$ 满足开头落在这个区间内部的最大重复子串的循环节个数都为 k 。只需要找字典序最小的一个。那么用 ST 表查一下这个区间内最小的 $rank$ 的后缀就好了。时间复杂度 $O\left(\sum_{i=1}^n \frac{n}{i}\right) = O(n \log n)$ 。有一个优化，当求得一个极长重复子串之后，落在子串内的 $s[i|L|]$ 都不用再枚举了。

2. **DONE BZOJ 4310 跳蚤** <2017-11-29 Wed>

后缀数组:ST 表: 二分查找

Description

给出一个字符串 S ，将它分成不超过 k 个子串，对于每个子串 T ，设 T' 为其最大子串，最小化选出的 k 个 T'_i 的最大值。

Solution

最大值最小可以二分。可以利用后缀数组求出本质不同的子串个数，对于一个第 k 大的子串，可以利用后缀数组求出它具体是谁。然后从后向前贪心扫描分块即可。注意一下比较两个串大小的细节。设第 k 大的子串和要比较的串在后缀数组中第一次出现的位置分别为 p_0, p_1 ，分 $p_1 < p_0, p_1 = p_0, p_1 > p_0$ 三种情况讨论即可。