Task

MiracleEEEE

December 11, 2017

Contents

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 鏼题计划

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 五 >* 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 五 >* 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 $s[0], s[|L|], s[2|L|], \cdots$ 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，所以还需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 $O\left(\sum_{i=1}^{n} \frac{n}{i}\right) = O(nlogn)$。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 三 >* 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时的细节。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。

2.2.2 2017 年 12 月 [2/2]

1. **DONE** BZOJ 3514 Codechef MARCH14 GERALD07 加强版 *<2017-12-06 三 >* LCT: 主席树: 贪心

   Description

   给出一张无向图，每次询问边标号在 $[l, r]$ 区间内的子图的联通块个数。$N \leq 200000$，强制在线。

   Solution

   考虑从特殊入手，如果是一棵给出的是一颗树，显然每次询问的答案是 $n-(r-l+1)$，图相比较树的区别是可能会出现环。构成环的边对答案是没有贡献的。如果我们能将每次询问对答案没有贡献的边都找出来，那么就解决了问题。按照编号从小到大依次加边，如果出现了一个环，不妨贪心的将编号最小的那一条边去掉（去掉最小的边不会影响答案 !），设最小的那条边的编号为 k，如果 k 在 $[l, r]$ 之间，那么现在加的这条边是没有用的，否则如果 $k < l$，那么这条边有用，对答案有 -1 的贡献。也就是说我们需要求出加入每一条边之后去掉的边是哪一条，可以用一颗 LCT 来维护。对于查询，也就是查区间内小于 l 的数字个数，直接上主席树就好了。

2. **DONE** Atcoder ARC086 D Non-decreasing *<2017-12-11 一 >* 构造

   Description

   给出一个序列 a_n，每次可以执行操作 $a_j += a_i$，构造方案使得操作次数 $\leq 2n$ 且数列单调不降。

   Solution

   不妨从特殊开始考虑，如果这个序列都为正数，那么显然只需要做一次前缀和就满足要求了。现在问题是数列不一定都是正数，那么只需要对每个数加上最大值即可（负数的情况类似讨论）。

3. **DONE** Atcoder ARC086 E Smuggling Marbles *<2017-12-11 一 >* 树形 DP: 队列: 计数

   Description

   有一棵树，根为 0，每个点可能会有一颗石子，重复进行如下操作：

   (a) 如果 0 号节点有一颗石子，那么把这颗石子放到盒子里
   (b) 把每个石子移项它的父亲节点
   (c) 如果有一个节点的石子数量超过 1 个，那么删除这些石子。

   问对于所有的 2^n 种起始情况，最后盒子里面的石子总数是多少。

   Solution

   每一层都是独立的，考虑枚举深度，用 $f[i][j](0 \leq j \leq 1)$ 表示节点 i 有 j 个石子的方案数，进行树形 DP 枚举它的哪个孩子给它 1 的贡献转移。但是这样是 $O(n^2)$ 的。

   但是其实每个动态规划的线程都是独立的, 不妨一起进行, 用双端队列维护状态 $f[i][j][k](0 \leq k \leq 2)$, k 的三维可以用一个三元组维护，表示相对结点 i 的深度为 j 的结点对 i 贡献 k 个石子的方案数（$k = 2$ 时表示有大于等于 2 个石子，这个状态是有必要维护的，这是与算法一的区别，因为转移的时候我们没有办法利用补集转移）。合并的时候就把小的队列与大的合并。只有深度相同的点合并才会对复杂度在 LCA 处产生贡献，而这些深度相同的点，设有 x_i 个，LCA 总数是 $x_i - 1$。那么从的复杂度为 $O(\sum x_i) = O(n)$。