Weekly Report 1

AsZ yuzhou627

April 11, 2015

Contents

1 CERC L

1.1 Discription

题目来源:$CERC\ L$[1]
有 n 个怪物会在 a_i 时刻出现在 d_i 处, 你必须在 b_i 时刻 (包含) 之前干掉他, 代价是 d_i, 但是你一旦选择放招, 就能干掉目前在 $0 \to d_i$ 范围内并且已经出现的所有怪物. 问干掉所有怪物最小代价.

[1] http://codeforces.com/gym/100543

1.2 Solution

1.2.1 Key observation

首先要转化成二维平面上的关系, 有很多平行 x 轴的线段, 左右分别时 a_i b_i, 高度是 d_i, 你可以画竖线, 保证和每条线段都有交, 问竖线长度之和最小值.
然后是要发现一个很关键的东西, 我们一定可以把竖线的位置定在某些 a_i 上, 这是因为任何不再 a_i 上的, 我们可以把他往左移动, 直到遇到另外的边界, 或者移到了这个线段的端点处, 就是 a_i 啦.

1.2.2 Details

知道了上面的结论, 就可以做区间 dp 了. 思路的启发: 你需要考虑最高的那根竖线, 然后决定了之后, 他就把所有的线段分成了左右两边 (当然出去了被这个竖线打掉的怪物了之后), 就是子问题了, 那就是区间 dp 了, 因此我们需要先将关键点全部存下来, 然后去重做区间 dp, 写成记忆化搜索会很简单.

1.2.3 Time Complexity

状态是 $O(N^2)$ 的, 转移是 $O(N)$, 因此是 $O(N^3)$.

1.2.4 Code

https://github.com/yuzhou627/ACM/blob/master/AsZ/2015.03.15/L.cpp

1.2.5 Mistakes

关键点的 $size$ 要用新的变量存下来, 而不能用 n, 会和怪兽数量的 n 冲突, 每次 dp 记得设置默认的值. 然后明确状态的含义, 是打掉完整包含在区间里面的线段的最小花费, 这一步需要枚举所有完整包含在区间里的线段, 直接枚举 $1 \to n$ 即可.

2 CERC G

2.1 Discription

给一个 DNA 序列, 需要从空串得道给的序列, 操作只能是在头或者尾加一个字符 A T G C, 或者将目前的序列反转之后接在前面或者后面, 问最少操作步数.

2.2 Solution

2.2.1 Key observation

如果我们能得到所有的偶数长度的不同的回文, 并且, 知道得到这个回文所需要的代价,dp 值, 那么我们就能做原题了, 答案上界是 n, 然后如果有偶数回文, 我们可以先构造出他的一半然后反转, 其他多余的部分用加 1 操作得到,(显然相同的回文只要考虑一次就可以了), 那么构造出一半的回文就是子问题了. 这题关

键在于求出构造出一半的回文的 dp 值. 于是这就是个字符串 dp. 而回文自动机可以做到求出所有回文字符串的同时得到它的 dp. 就做完了.

2.2.2 Details

回文自动机.

2.2.3 Time Complexity

$O(N)$

2.2.4 Code

https://github.com/yuzhou627/ACM/blob/master/AsZ/2015.03.15/G.cpp

2.2.5 Mistakes

这题细节在于一半的回文的 dp 的求解.

3 NEERC Sub A

3.1 Discription

题目来源:$NEERC\ Sub\ A$[2]
给出 n 条边, 及 q 个询问, 查询区间 $l \to r$ 之间的边的导出子图是否是二分图.(可以黑白二染色, 这里用奇圈刻画的话, 没有效果, 不好理解描述, 虽然是一个东西).

3.2 Solution

3.2.1 Key observation

没有修改, 直接回答所有询问, 莫队算法, 分块解决.

3.2.2 Details

考虑分块之后, 莫队的算法会将询问分成两部分, 右边的每次加一条边, 就是并查集的更新, 我们直接要对当前的每个块给出一种染色方案, 如果遇到冲突, 直接调整, 为了能很快调整出来, 我们需要记下每个并查集的集合里面的点的标号. 如何判断合法呢? 两个合法的联通块合并肯定是合法的, 但是可能连接处同色了, 这时候就把小的那个联通块颜色反转一下. 否则就是在一个块内连边, 这时候就看颜色是否冲突了, 冲突就跪了. 再考虑左边的那些 $O(\sqrt{N})$ 的边如何处理. 因为这些边是每个询问不公用的,(右边那些是公用的), 那么我们除了需要维护出答案, 还需要回到历史状态, 这时候就需要两个并查集了, 一个 f 保存只有右边的边的操作的状态, 一个 g 保存了当前 f 加上左边那些边之后的状态, 那么处理完这些之后, 如何复原, 再把所有的使用了的边处理一遍, 但是这次是在 f

[2] http://codeforces.com/gym/100513/problem/A

里面查祖先, 并且把 g 里面 x 对应的点的祖先修改换成 f 中 x 对应的祖先, 并且把 g 里 x 所在那个联通块集合也复原成 f 那样的. 这里实际上只需要把 $vector$ 末尾超出的所有元素弹出去就可以了. 这题就做完了.

3.2.3 Time Complexity

$O(Q * logN * \sqrt{M})$, 其中分块是根号, 但是启发式合并还带来了一个 $O(logN)$.

3.2.4 Code

https://github.com/yuzhou627/ACM/blob/master/AsZ/2015.03.22/A.cpp

3.2.5 Mistakes

写代码的时候, 注意这里莫队是对所有边分块, 所以排序的时候参数是 m 不是 n, 这种细节一定要注意. 有可能看很久的.$init$ 的时候 set 要记得清空. 写莫队的时候, 要记得如果左端点进入新的一个块了, 那么要把 L R 设置成默认的值. 另外, 有可能有些询问的左右端点都在一个块里面, 也就是莫队的左右根本没有右边那块, 这个时候右边加边的起点一定要设成上一次的 R 和左右分界点的较大值,(因为默认把 R 设置成了 $-\infty$, 如果没有更新过 R 的话就会 RE). 另外做完了询问, 别忘了把 L R 设置成当前询问的 l r.L 的作用只有判断询问的左端点是否进入了一个新的块, 即是否需要重新初始化了.

4 Camp Day3 C

4.1 Discription

一个连用无向图, 有 k 个关键点, 求出每个关键点距离最近的那个关键点.

4.2 Solution

4.2.1 Key observation

可以用 $Dijkstra$ 算法来做.

4.2.2 Details

考虑原来的最短路算法的关键就在于松弛操作, 那么现在我们也可以做松弛. 维护和每个点最近的两个关键点的距离以及点的标号. 那么可以用当前的结果去对其他点做松弛. 算法一定会结束. 一开始先将每个关键点的最近关键点设成自身, 并且扔进堆里, 然后显然我们应该用距离最小的状态来松弛其他节点. 但是这里要注意一些松弛的细节. 详见代码.

4.2.3 Time Complexity

$O(NlogN)$

4.2.4 Code

https://github.com/yuzhou627/ACM/blob/master/Camp/2015.02.03/C.cpp

4.2.5 Mistakes

这题我写错了很多地方.

5 Project Euler 500

5.1 Discription

求最小的含有 2^{500500} 个约数的数, 取模.

5.2 Solution

5.2.1 Key observation

如果 $n = p_1^{a_1}...p_k^{a_k}$, 那么 $\sigma(n) = \Pi_{i=1}^{k}(a_i + 1)$.

5.2.2 Details

这题我的方法本来是不会证明的,(有点贪心, 不加证明的意思), 后来看了别人的做法之后, 觉得是对的, 但是复杂度更好. 但是不太好想到和理解.
首先, 我们知道了最后的答案是 $n = p_1^{2^{a_1}-1}...p_k^{2^{a_k}-1}$ 的形式. 而且满足 $\sum_{i=1}^{k} a_i = 500500$. 那么我们就是需要最小化答案, 考虑调整法. 现在是和恒定, 那么如果我单独考虑两个质数因子:

$$A = p_1^{2^a-1}p_2^{2^b-1} \ p_1 < p_2$$

调整一下, 把 a 调大 1 的同时保证和不变, 调小 b.

$$A' = p_1^{2^{a+1}-1}p_2^{2^{b-1}-1} \ p_1 < p_2$$

于是我们比较两个数子大小就知道应该如何调整方案了. 于是, 我们可以一开始把前 500500 的质数的幂次都赋予 1. 然后不断调整, 但是如果每两个数都两两比较一下, 然后决定是否调整的话, 复杂度是 $O(N^2)$ 的, 但是我这里不加证明得觉得, 应该是从前往后, 不断调整, 用最小的质因子去换最大的质因子, 然后维护指针就好了. 这样确实也得到了正确答案. 所以复杂度就是 $O(N)$ 的.
我们换一个角度思考问题: 因为这题最后答案总是 $n = p_1^{2^{a_1}-1}...p_k^{2^{a_k}-1}$ 的形式, 我们考虑下式:

$$p^{2^a-1} = p^1p^2p^4...p^{2^{a-1}}$$

所以我们每次调整 $a+1$ 和 $b-1$ 的操作, 实际上就是在把第一个质因子的最后那个 $p_1^{2^a}$ 部分加上, 而去掉了第二个质因子的最后那部分 $p_2^{2^{b-1}}$. 所以就是用一个数换另一个数的过程, 于是嘛, 我们把所有可能用到的数, 长成 $p_k^{2^k}$ 的数都存下来, 取最小的 500500 个就好了. 考虑到最大质数 p_{500500} 的次数是 1, 所以其他的数都不能比他大. 所以总共不会有太多的数啦. 这样就能说明我前面的做法是对的了. 但是是 $O(Nlog(N))$ 的.

5.2.3 Time Complexity

$O(N)$

5.2.4 Code

https://github.com/yuzhou627/ACM/blob/master/Euler/P500.cpp

5.2.5 Rererence

https://oeis.org/A050376

5.2.6 Mistakes

下标别搞错就好了.

6 Project Euler 509

6.1 Discription

三堆石子的变形 Nim 游戏, 每堆石子数 $\leq n$, 问后手必胜的石头分布方案数.

6.2 Solution

6.2.1 Key observation

打表, 发现这个游戏十分特殊,sg 值是石子数的二进制右起连续的 0 的个数,(就是满足 $2^k|n$ 的最大 k).

6.2.2 Details

那么在给定范围内,sg 值的范围只有 50. 于是我们直接统计出每种的数量, 然后枚举两个, 直接乘另一个就好了.

6.2.3 Time Complexity

$O(50^2)$

6.2.4 Code

https://github.com/yuzhou627/ACM/blob/master/Euler/P509.cpp

6.2.5 Rererence

http://oeis.org/A007814

6.2.6 Mistakes

不要忘记打表, 锻炼观察能力, 不借助 $oeis$ 也要能发现规律.

7 SPOJ QTREE

7.1 Discription

题目来源: $SPOJ\ QTREE$ [3]
给一棵树, 两个操作, 修改边的权值, 询问 $a \to b$ 路径上的边权最大值.

7.2 Solution

7.2.1 Key observation

树链剖分入门.

7.2.2 Details

树链剖分回顾.
树上的任何路径都能被分成重链和轻链, 每种都是 log 段. 每个点有个 son 表示他的重儿子标号. 还有最重要的一个数组 pos 表示每个点对应的点或者边在线段树里面的位置.pre 数组是必要的, 因为常常需要知道一个点的父亲的标号,$size$ 数组是辅助求 son 数组这些东西的.

7.2.3 Time Complexity

$O(Nlog(N)^2)$

7.2.4 Code

https://github.com/yuzhou627/ACM/blob/master/SPOJ/QTREE.cpp

7.2.5 Mistakes

实现并不难, 稍微有些细节, 尤其是这里的 $find$ 函数博客作者实现得很好.

7.2.6 Reference

http://blog.sina.com.cn/s/blog_6974c8b20100zc61.html

[3] http://www.spoj.com/problems/QTREE/