# 第 23 届全国青少年信息学奥林匹克联赛模拟题解

# CCF-NOIP-2018

## 提高组(复赛)第二试

### 竞赛时间：2018 年 11 月 5 日 8:00-11:30

| 题目名称 | 凯旋而归 | 生死之境 | 走向巅峰 |
|---|---|---|---|
| 题目类型 | 传统型 | 传统型 | 传统 |
| 目录 | ak | life | winer |
| 可执行文件名 | ak | life | winer |
| 输入文件名 | ak.in | life.in | winer.in |
| 输出文件名 | ak.out | life.out | winer.out |
| 每个测试点时限 | 1 秒 | 2 秒 | 1 秒 |
| 内存限制 | 512MB | 512MB | 128MB |
| 测试点数目 | 10 | 10 | 20 |
| 每个测试点分值 | 10 | 10 | 5 |

提交源程序文件名

| | | | |
|---|---|---|---|
| 对于 pascal 语言 | ak.pas | life.pas | winer.pas |
| 对于 C 语言 | ak.c | life.c | winer.c |
| 对于 C++语言 | ak.cpp | life.cpp | winer.cpp |

编译选项

| | | | |
|---|---|---|---|
| 对于 C 语言 | -lm | -lm | -lm |
| 对于 C++语言 | -lm | -lm | -lm |

注意事项：

1、 文件名（程序名和输入输出文件名）必须使用小写。
2、 除非特殊说明，结果比较方式均为忽略行末空格及文末回车的全文比较。
3、 C/C++中函数 main()的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
4、 全国统一评测时采用的机器配置为：CPU AMD Athlon(tm)II x2 240 processor，2.8GHz，内存 4G，上述时限以此配置为准。
5、 只提供 Linux 格式附加样例文件。
6、 评测在 NOI Linux 下进行。
7、 编译时不打开任何优化选项。

# 1. 凯旋而归

**算法 1**

记录前缀异或和，根据题目定义$O(n^2)$计算帅气值，期望得分$50$.

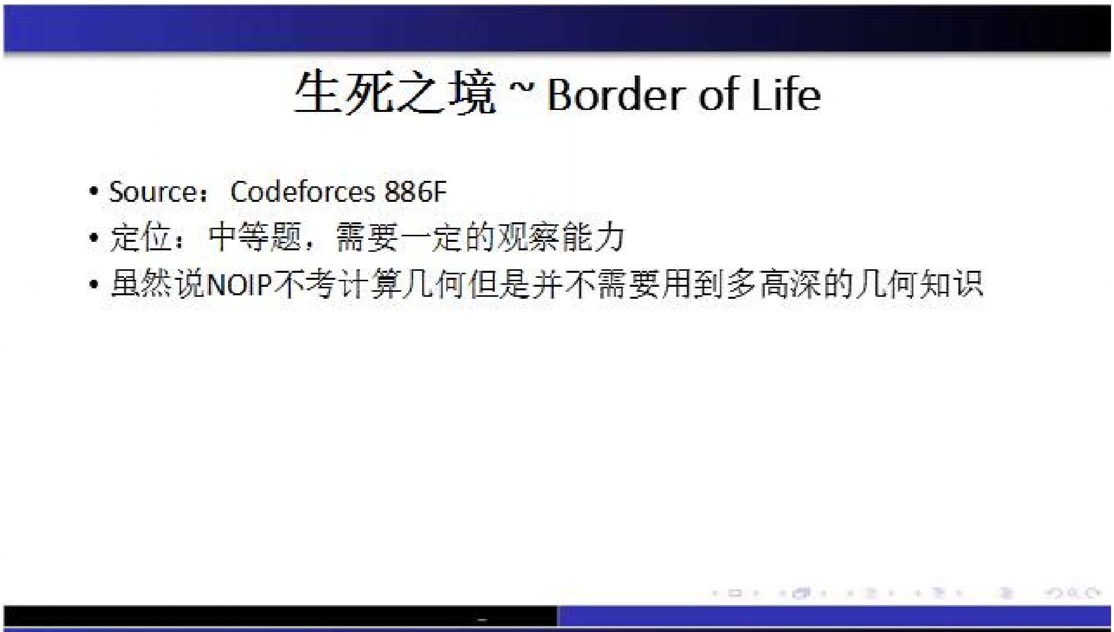
**算法 2**

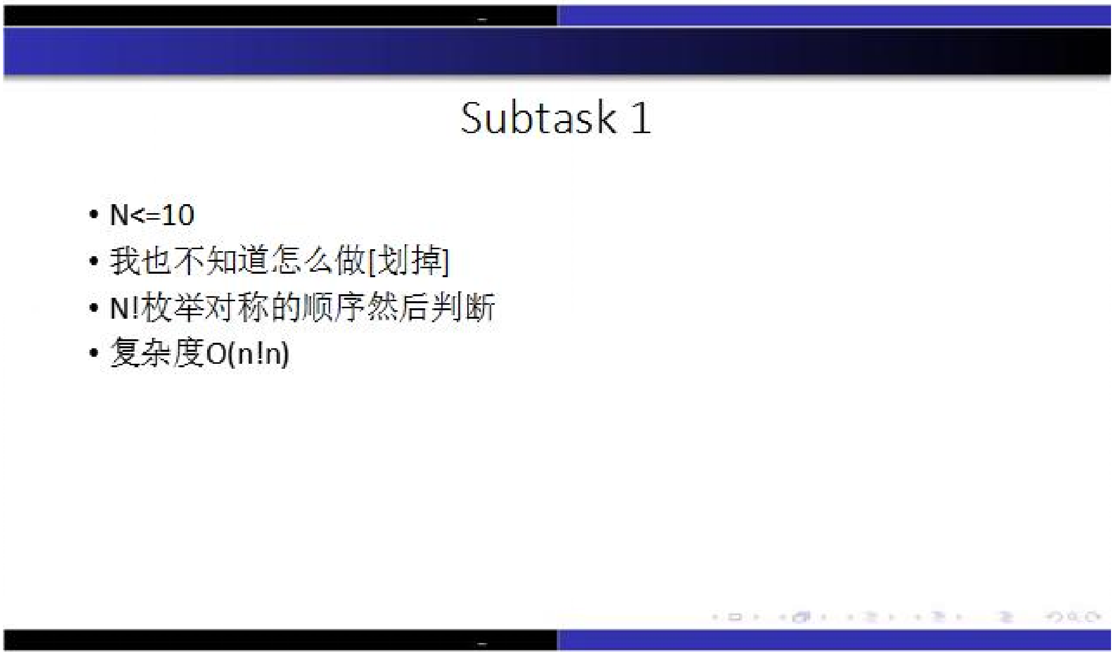
记$a$序列的前缀异或和为$b$，那么对于每个$i$，我们便是要求解一个$j(0 \le j \le i)$使得$b_i\ \ xor\ \ b_j + b_j$最大。

现在一个显然的结论就是如果$b_i$在二进制下的某位是$1$的话，那么$b_j$在该位的值不会对答案贡献有任何的影响，也就是说我们只需要关心$b_i$在二进制下是$0$的位。

记$f_i$表示满足$i\ \ and\ \ b_j = i$的最小的$j$，现在从高位往低位贪心，记当前找到的$b_j = now$，我们现在想把第$k$位的贡献从$0 \times 2^{k-1}$变成$2 \times 2^{k-1}$，只需判断$f_{now\ \ xor\ \ 2^{k-1}}$是否小于等于 i，若是，则$now = now\ \ xor\ \ 2^{k-1}$,贡献便多了$2 \times 2^{k-1}$，否则不作修改。时间复杂度$O((n + max(a_i))\, log\, max\,(a_i))$，期望得分$100$。

# 2. 生死之境


生死之境 ~ Border of Life
• Source： Codeforces 886F
• 定位： 中等题，需要一定的观察能力
• 虽然说NOIP不考计算几何但是并不需要用到多高深的几何知识


Subtask 1
• N<=10
• 我也不知道怎么做[划掉]
• N!枚举对称的顺序然后判断
• 复杂度O(n!n)

## Subtask 2

- N<=500
- 根据人类智慧我们可以发现，对称中心一定是原点集的重心
- 枚举两个点，令其对称，可以得到O(n^2)条直线，显然只有这些直线可能成为答案
- 有了上面的性质判断一次是O(n)的
- 复杂度O(n^3)

## Subtask 3

- N<=2000
- 首先把原本就关于重心对称的点删去
- 如果剩余的点数<=2就是-1
- 否则我们找出这O(n^2)条直线
- 注意到一条直线要合法必须出现>=n/2次
- 这样的直线最多O(n)条，暴力判断即可
- 复杂度O(n^2 log n^2)

# 3. 走向巅峰

# 题目大意

对于一棵树，每次随机染黑一个叶子（可能会重复染黑），期望多少次后直径变小

## Solution

先考虑直径 R 为偶数的情况：
这种情况下，显然可以找到一个点 root，使得所有的直径都经过它，以这个点为根给每个点定深度 $dp_x$，
那么，只有 $dp_x = R/2$ 的点才有可能为直径端点，剩下的 $dp_x < R/2$ 的叶子为无关点，先统计出来设有 m1 个，
把所有 $dp_x = R/2$ 的点按他是 root 的哪一棵子树分成几个集合，
直径改变了，当且仅当只剩下一个集合的点没有被删完，（染黑）

对于 R 为单数的情况：

显然有必经边，那么就以这条边切开两半，也就是只有两个集合，集合中的点为 $dp_x = \lfloor \frac{R}{2} \rfloor$，m1 也一样统计，这样就和偶数的一样了

有一个值是可以先预处理的：可以推出，当全局还剩 x 个叶子没被删时，再删掉一个没被删的点的代价为$\frac{m}{x}$，m 为全部叶子数，

对于无关点，我们可以视作，这些点已经被删掉了，也就是一开始就已经删掉 m1 个点，

那么现在问题就转化成：每次删掉一个没有删掉的点（带权），求删剩一个集合的期望，
这个可以用(所有方案代价总和)/(方案数)的方法算概率，

枚举一个集合（大小为 d），假设最后剩下它，其他的集合全选完，再枚举这个集合最后选了 i 个，贡献为：（d0 为所有集合大小）

$$\sum_{i=0}^{d-1} C_d^i * (d0 - d + i - 1)! * (d0 - d) * (\sum_{j=d-i+1}^{d0} \frac{m}{j}) * (d - i)! * \frac{1}{d0!}$$

（注意：要保证最后一个选的一定不是当前集合的点，要不会算重）
因为无限次的染色一定会全部染上黑色，后边的(d-i)!表示剩下的乱选，d0!表示全部的方案。

复杂度：$O(n\log(n))$（计算逆元要个 log）