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，根据题目定义计算帅气值，期望得分.

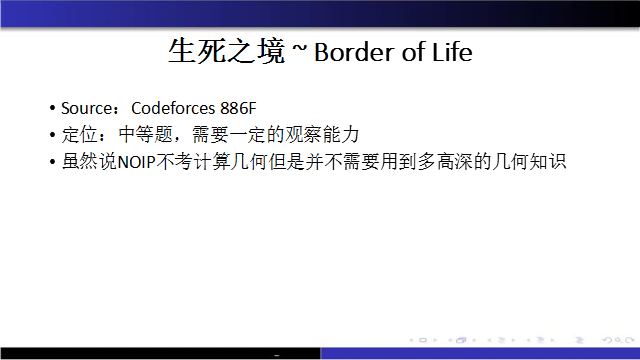
**算法2**

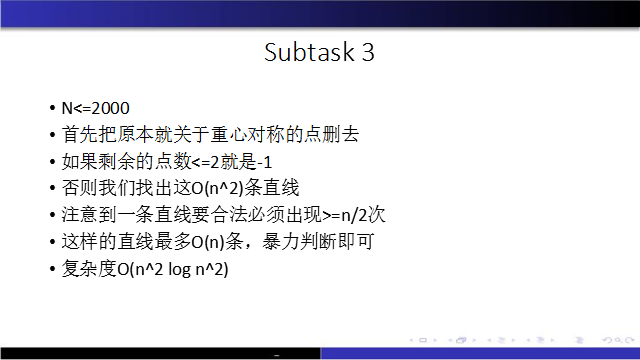
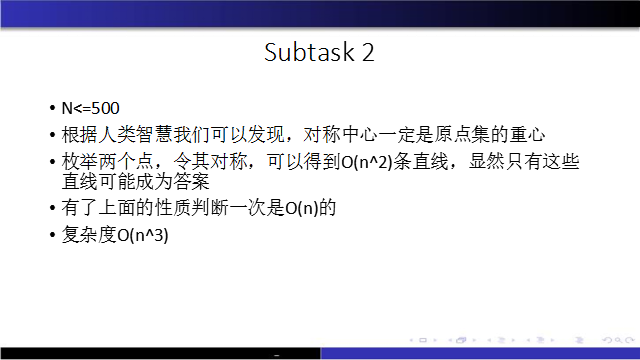
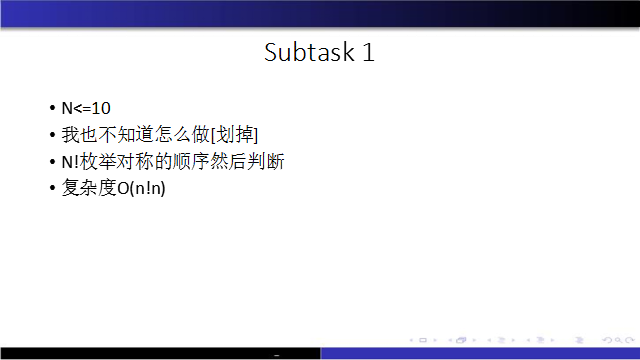
记序列的前缀异或和为，那么对于每个，我们便是要求解一个使得最大。

现在一个显然的结论就是如果在二进制下的某位是的话，那么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在二进制下是的位。

记表示满足的最小的，现在从高位往低位贪心，记当前找到的，我们现在想把第位的贡献从变成，只需判断是否小于等于i，若是，则,贡献便多了，否则不作修改。时间复杂度，期望得分。

2.生死之境





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，

那么，只有的点才有可能为直径端点，剩下的的叶子为无关点，先统计出来设有m1个，

把所有的点按他是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，m1也一样统计，这样就和偶数的一样了

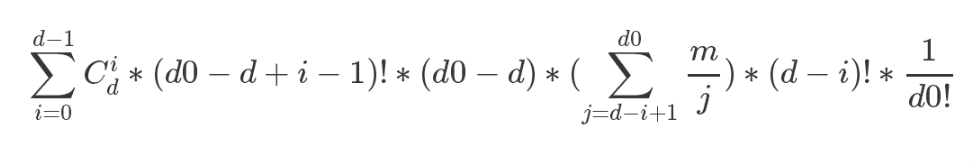
有一个值是可以先预处理的：可以推出，当全局还剩x个叶子没被删时，再删掉一个没被删的点的代价为，m为全部叶子数，

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）



（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：（计算逆元要个log）