FFC2077解析

1. 不解释
2. B 显卡，CD 是 CPU，A 是我瞎编的
3. A是在 window 控制台的命令，B 是g++编译，C是我瞎编的
4. 计算器
5. 7 个点最多 21 条边，去除树的 6 条边显然不行，同理可得最小有 8 个点
6. 分别计算表达式的值即可，发现 BCD 均为 `FALSE` 而 A 为 `TRUE`
7. 枚举 ξ 分别为 $1,2,3$ 的概率，即

   $$P(\xi=1)=\frac{C(3,1)*C(10,3)}{C(13,4)}=\frac{360}{715}$$

   $$P(\xi=2)=\frac{C(3,2)*C(10,2)}{C(13,4)}=\frac{135}{715}$$

   $$P(\xi=3)=\frac{C(3,3)*C(10,1)}{C(13,4)}=\frac{10}{715}$$

   $$E(\xi)=P(\xi=1)\times 1+P(\xi=2)\times 2+P(\xi=3)\times 3$$

   $$E(\xi)=\frac{360}{715}+\frac{270}{715}+\frac{30}{715}=\frac{660}{715}=\frac{660}{715}=\frac{132}{143}=\frac{12}{13}$$
8. 枚举选了几个赝品，除以总方案数，答案是

   $$\frac{1\times C_3^1\times C_{10}^3+2\times C_3^2\times C_{10}^2+3\times C_3^3\times C_{10}^1}{C_{13}^4}$$
9. 转换为前缀表达式为 `+*/abc-de` ,然后转化出中缀表达式 `a/b*c+(d-e)` 算出值为 `-1`
10. 由于 $T(n)=3T(\frac{n}{4})+n\log_2 n$

    我们可以根据主定理的形式， $T(n)=aT(\frac{n}{b})+f(n)$

    那么带入得到 $a=3,b=4,f(n)=n\log_2 n$

    那么基准函数为 $\Theta(n^{log_4 3})\approx\Theta(n^{0.8})$

    而 $f(n)=n\log_2 n$ 显然比 $\Theta(n^{0.8})$ 强，$\varepsilon=0.1$ 时依旧比 $n^{0.8}$ 大

    再看看是否满足 $af(\frac{n}{b})\le cf(n)$

    得到 $\frac{3}{4}n\log_2\frac{n}{4}\le\frac{3}{4}n\log_2 n$

    $\therefore\exists c<1$ 所以 $T(n)=\Theta(f(n))=\Theta(n\log_2 n)$
11. 生成函数，不过可以分类讨论（cdx表演！）
12. 模拟即可。
13. 这题主要考验了选手是否会 `Kruskal`，首先进行一次边的排序，复杂度 $O(n)$

    然后枚举每条边，每次都要查询两个端点，个别的会进行合并

    总的复杂度应该为 $O(n+m\alpha+n\alpha)\to O(m\alpha)$

14. 模拟一下 dij 求最短路方案数的过程。

15. 先手赢的情况不存在，证明：

> 如果你拿先手必胜，设你最优策略是拿第 i 行的第 j 个子。
>
> 现在你换后手，那么另一个人开局策略有：
>
> 1. 不拿第 i 行。就当成是你先拿了第 i 行的第 j 个然后它拿了这个位置，你正常跟它博弈。
> 2. 拿第 i 行 $<j$ 列。就当成是你先拿了第 i 行的第 j 个然后它拿了这个位置。
> 3. 拿第 i 行 $>j$ 列。你直接拿走第 i 行第 j 个，相当于你先手。
>
> 对方不可能拿第 i 行第 j 列，因为颜色不一样。
>
> 也就是说，你拿先手必胜，则你拿后手也必胜。

那么考虑下面这个局面

```
1111
00
0
```

显然 A 赢对吧，那我们就可以找到一个思路，一个黑棋是 1 贡献，一个白棋是 -1 贡献。

不难得知，最终贡献 >0 就是 A 赢，贡献 <0 就是 B 赢，贡献 $=0$ 就是后手赢。

然后考虑下面这个：

```
10
```

单独拎出来，显然 A 必胜，因此 `10` >0。又因为

```
10
0
```

B 必胜。因此 `10` <1。

猜测 `10` $=\dfrac{1}{2}$，可以用这个：

```
10
10
0
```

后手赢，因此贡献为 0，所以 `10` $=\dfrac{1}{2}$。

同理有 `100` $=\dfrac{1}{4}$，`101` $=\dfrac{3}{4}$ 等等。

又发现把开头的字符复制一个贡献会 $+1$，比如 `1100` $=$ `100` $+1=\dfrac{1}{4}+1=\dfrac{5}{4}$。原因是拿的时候必然只会拿连续段中靠后的，这样就不会白白消掉自己的棋子。

所以就可以得到贡献是 $(-1)+(2+\dfrac{1}{2})+\dfrac{1}{4}+(-\dfrac{3}{4})+(-\dfrac{1}{2})+(-\dfrac{1}{2})=0$，所以后手必胜。

阅读程序 T1

简单的矩阵乘法。

16. 就算是 0 后面乘法也没影响
17. 所有矩阵的 n, m 全都是 4
18. $n = 1$ 快速幂死循环
19. 算的是同一个东西，见第 21 题
20. 手算 10 项不难吧（$2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123$
21. 显然。

阅读程序 T2

求的是这个东西：

$$\sum_{l=1}^{n}\sum_{r=l}^{n} Max(l,r) \bigoplus Min(l,r)$$

Max 表示 a_l 到 a_r 的最大值，Min 同理，$\bigoplus$ 是异或。

算法是 cdx 分治，c 维护的是插入一个数，删除一个数，查询所有数和给定数的异或和。

22. 显然会影响
23. 考优先级，括号可以删
24. 数据范围只有 1000，$\log$ 小于 16，不会影响
25. 60 行会执行，56 行不会
26. 分治 $n \log n$，c 数组还有一个 $\log$
27. 稍微算一下。

阅读程序 T3

28. 第一行和第二行都为第二类斯特林数，但是第二行的在 $s2[i-1][j] * j$ 处未取模，所以会出现负数，**错误**。
29. `func1` 代表的是相当于给定 n 个相同的球，放进 m 个不同的盒子，不可以有空盒的方案数。而非可以有空盒，**错误**。
30. `func3` 和 `func4` 由 `auto` 推导，后接 `lambda表达式` ,返回值和两个参数均为 `int` 故类型为 `std::function<int(int, int)>` ，**正确**。
31. `s` 为第二类斯特林数，**错误**。
32. 直接模拟即可，选择 `C`。
33. 数据范围较小，直接手算，枚举 $m \in [1, 4]$ 算出其最大值为 7, 选择 `D`。

完善程序T1

很多人可能都不会树状数组求区间 max，实际上是可以做到的。

单点修改的时候多加一个数组维护每一个点的值。

查询 l 到 r 的最大值时候分情况：

- 如果 c_r 维护的区间，也就是 $r - lowbit(r)$ 到 r 的左端点 $r - lowbit(r)$ 在 l 右边，直接正常跳即可。
- 否则将 r 减一，r 的这一位用多开的数组直接更新答案。

完善程序T3

rt 是所有询问点在以宝藏位置为根的 LCA，s 是宝藏。

44. 答案最大值是 n，一共 $n \times n!$ 个方案，所以初值是 $n^2 \times n!$
45. 可以确定 rt 是所有询问点在以宝藏位置为根的 LCA 的条件：（rt 被选或者 rt 的至少两个子树里有点被选）并且 rt 的宝藏所在的子树里没有点被选。
46. 如果 s 和 rt 相同，只要 rt 被选或者 rt 的至少两个子树里有点被选就可以。
47. 如果 fas 的子树里有两个或以上和 s 深度相同的，即使知道 rt 是 LCA 也没法判断。
48. 如果 rt 的某个儿子的子树里有和 s 的深度相同的（且 s 不在这棵子树中）时这个子树里必须选点。

如果要详细的解析，我可以讲评的时候讲一讲（