

思路

- 给5分钟思路思考时间，若没有想法，那么就可以考虑看解析，后面再自己完成编程。

数学运算

- 对一些 n 阶或 n 个的数学相关的题，可以按顺序先算一些出来，**找找是否有规律**。若有规律，利于解题，也有利于优化运行时间。（遇到好多次斐波那契数列）
- 对于数学计算题（阶层或二进制有关的运算），考虑使用位操作位（ \ll ， \gg ）来简化运算次数。 **\wedge 是异或运算**，两个数的二进制形式，按位对比，相同取0，不同取一。
- 关于 $\&$ 还有 \parallel 运算的细节。例如 `return f1() || f2();` 如果 `f1()` 函数返回值为 `true` 的话就不会运行 `f2()` 函数。编程的时候语句顺序很重要。

树

- 对于层次遍历的问题，可以用队列去完成这个功能。（比如树的第二层遍历过程中，第三层就按顺序入队了）
- 中序遍历的非递归方法，使用栈。

数组

- `sort` 函数完成排序
- 字符串数组局部替换问题
确定替换后的字符串数组长度，再从后往前逐个替换。

链表

- 拆分链表时
少用 `->next->next`，用两个指针遍历实现。
- 遍历链表的指针 `p`
通过判断 `p` 是否为 `NULL` 来判断结束。（`p->next` 来判断易出错，原因是遍历的时候 `p = p->next`，会跳过两个对象）
- 对于新的链表指针
第一步先加入一个节点，后续再使用循环完成操作。

特别需要注意的地方

- 对于程序输入的值要进行判断，不符合要求时即需要返回指定的数。要把判断情况写全。
- 编程过程中，凡是有返回值的，都要进行判断，有不同的处理结果。