信息学奥赛（NOIP）复赛学习方法推荐

一、确定你的语言

NOIP包括三种语言c/c＋＋/pascal，在最初必须确定自己使用的语言。没有c/c＋＋基础的，个人建议使用pascal，因为它更容易上手，如果有充裕的时间，则建议c/c＋＋，因为它们对你今后的程序编写，更有益处。

二、从排序入手

排序是基础中的基础，快速排序是必备本领，方法就是背下来。c/c＋＋是自带快排的，因此很轻松。多关键字排序和稳定排序也是必须掌握的排序知识。

三、贪心和穷举以及模拟——最简单的程序

想得奖，必须掌握贪心和穷举以及模拟，虽然不能让你得满分，但可以给你拿到30-60分。它们是你想不出更好算法时的救命稻草。

贪心算法（又称贪婪算法）是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，他所做出的是在某种意义上的局部最优解。但是贪心是可以得分的。

枚举算法是指，列举出所有可能的取值，从中找出最优解。

模拟算法是指，通过逐步进行操作、逐步判断来推断是否符合题目中所给出的情况。非常耗时，一般不可能得到最优解，但是可以得到部分分数。

四、用动态规划来训练思维

比较难，对思维的周密程度和逻辑要求非常高。可以用来训练思维，对于学习时间短的筒子，动态规划可以帮助你迅速进入编程状态，也有助于帮你发现题目背后可能隐藏的更简便的算法。

动态规划主要的思考规律应该如下：  
定义函数（动态转移方程中转移量的定义）——>建立方程——>确定初值和边界  
提醒！考场上想不到动态转移方程，请选择贪心、枚举或模拟等方法来获得部分分数。动态规划最后得出的答案不正确时，也不要耗费大量时间来找出错误，因为这非常难，也非常耗时间，得不偿失。

五、学习简单的图论

包括：（单源或多源）最短路和（最小）生成树。  
最短路中需要学习Dijkstra算法和Floyd算法。近年来图论题目越来越难，知识点越来越多，所以时间不够，请掌握这两种。

最小生成树需要掌握Prim算法和Kruskal算法。前者适用于稠密图，后者适用于疏密图。两者可以比较学习，看到它们的优点和不足。

六、常用的数据结构——让程序更快一点

最常用到的是堆（优先队列）、并查集以及树状数组堆。  
堆：只关注“直系亲属关系”，不关注“旁系”。常配合贪心使用。  
并查集：快速判断两个元素是否有关联，增加其他算法，还可判断元素间关系。  
树状数组堆：平衡查询和修改的操作复杂度的一种算法，常用于解决需要查询和修改的问题。  
七、搜索——和枚举很像

深度优先搜索和广度优先搜索。

深度优先搜索：一条路走到底。

广度优先搜索：每一步将下一步的可能性放入队列中，然后按照队列顺序来探测。

复赛中往往会加入很多复杂的元素，所以也需要好好掌握。  
八、最后列一下一定要学习的数学基础知识

快速幂、高精度、筛法选素数、辗转相除法