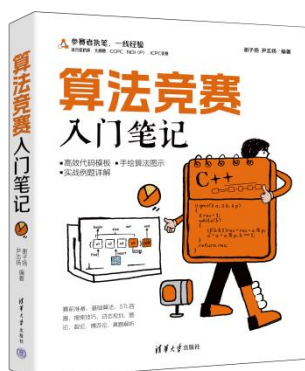


《算法竞赛入门笔记》学习指南



第 1 章 赛前准备学习指南

1.1 算法竞赛简介

目标：了解各类算法竞赛的背景和特点。

内容：

ACM-ICPC、CCPC、NOIP/NOI/CSP-J/S、蓝桥杯、天梯赛等竞赛简介。

学习方法：

- 阅读各竞赛的官方网站和相关资料，了解其历史、规则 and 影响力。
- 观看赛事视频或参与在线讲解，感受竞赛氛围。

1.2 语言和工具

目标：熟悉竞赛常用语言和编程环境。

内容：

- 竞赛语言（如C++）的特点和优势。
- 编程环境的选择和配置。
- 训练平台的使用方法。

学习方法：

- 安装并熟悉所选编程语言的编译环境。
- 在训练平台上进行基础编程练习，熟悉平台功能。

1.3 能力要求和学习建议

目标：制定个人化的学习和训练计划。

内容：

- 如何开始算法竞赛的学习。
- 高效训练的方法和策略。
- 补题和总结的重要性。
- 正确看待算法竞赛的付出与收益。

学习方法：

- 设定明确的学习目标和计划，分阶段提升能力。
- 定期回顾和总结，及时调整学习方法和策略。
- 保持积极心态，享受学习过程，面对挑战不气馁。

第 2 章 基础语法学习指南

2.1 第一个程序：Hello World

目标：掌握基础编程结构。

内容：

- 编写第一个程序：“Hello World”。
- 理解头文件、命名空间和main函数的作用。

学习方法：

- 跟随教材或在线教程，逐步编写并运行简单的程序。
- 分析每行代码的功能，理解其背后的逻辑。

2.2 输入与输出

目标：学会处理基本的输入输出操作。

内容：

- 使用scanf、printf进行C风格输入输出。

- 使用cin、cout进行C++风格输入输出。
- 通过示例掌握各种输入输出技巧。

学习方法：

- 完成相关习题，实践不同输入输出场景。
- 比较不同输入输出方法的优缺点，选择适合的场景使用。

2.3 常用的基础数据类型和数学运算

目标：熟悉基础数据类型和数学运算。

内容：

- 基本数据类型（如整型、浮点型）的使用。
- 常用的数学运算符和表达式。

学习方法：

- 通过实例学习不同数据类型的特性和用途。
- 解决简单的数学问题，巩固数学运算技能。

2.4 分支语句

目标：掌握条件判断和分支结构。

内容：

- if语句的使用。
- 三目运算符的应用。

学习方法：

- 编写条件判断程序，模拟实际场景进行决策。
- 优化代码结构，提高可读性和效率。

2.5 循环语句

目标：熟悉循环结构的使用。

内容：

- for循环的实现和应用。
- while循环的实现和应用。

学习方法：

- 完成循环相关的编程题目，加深理解。

- 分析循环性能，选择适当的循环结构。

2.6 数组

目标：掌握数组的基本操作。

内容：

- 数组的结构定义和初始化。
- 数组与指针的关系。

学习方法：

- 通过实例学习数组的声明、初始化和访问方法。
- 理解数组内存分配原理，优化数组使用。

2.7 函数

目标：学会函数的定义、调用和Lambda表达式。

内容：

- 函数的声明、实现和调用方法。
- Lambda函数的使用场景和优势。

学习方法：

- 编写多个函数，实现不同的功能模块。
- 理解Lambda表达式的原理，适时使用简化代码。

2.8 结构体

目标：掌握结构体的定义和使用。

内容：

- 结构体的定义和初始化方法。
- 结构体数组的实现和应用。

学习方法：

- 定义结构体变量，模拟实际问题中的数据结构。
- 操作结构体数组，实现数据的批量处理。

2.9 推荐代码规范

目标：遵循良好的编程习惯和规范。

内容：

- bits/stdc++.h头文件的使用。
- std命名空间的使用建议。
- 代码缩进、换行和循环规范。
- longlong类型的使用建议。
- 避免过度压缩代码和滥用宏定义。
- 适当撰写注释提高代码可读性。

学习方法：

- 阅读并模仿优秀的代码示例，逐渐形成自己的编程风格。
- 定期检查并更新代码，确保符合规范要求。

第3章 基础算法学习指南

3.1 时空复杂度分析

目标：掌握算法的时间和空间复杂度分析方法。

内容：

- 时间复杂度分析的基本概念和方法。
- 空间复杂度分析的基本概念和方法。

学习方法：

- 通过分析简单算法（如排序、搜索）的复杂度来理解基本概念。
- 实践计算不同算法的时间和空间复杂度，巩固理论知识。

3.2 暴力枚举

目标：理解并应用暴力枚举思想解决简单问题。

内容：

- 解空间的定义和理解。
- 枚举方法的应用。
- 通过例题讲解加深理解。

学习方法：

- 动手实现简单的暴力枚举算法，如全排列、子集生成等。
- 分析暴力枚举算法的优缺点，理解其适用场景。

3.3 二分法

目标：掌握二分法的原理和应用。

内容：

- 二分法的特征和类型。
- 通过例题讲解加深理解。

学习方法：

- 理解并实现标准的二分查找算法。
- 探索二分法在不同问题（如找极值、判断存在性等）中的应用。

3.4 双指针

目标：理解双指针技术的原理和应用。

内容：

- 双指针题的特征和类型。
- 通过例题讲解加深理解。

学习方法：

- 通过实例学习双指针的设置和移动策略。
- 尝试用双指针解决实际问题，如字符串处理、数组问题等。

3.5 其他

目标：了解并掌握其他基础算法思想和技术。

内容：

- 递归的基本概念和应用。
- 排序算法的基本思想和实现。
- 位运算的基本技巧和应用。
- 贪心算法的基本思想和应用场景。
- 分治法的基本思想和应用案例。

学习方法：

- 分别学习和实践每种算法和技术，理解其核心思想和应用场景。
- 通过解决相关问题来巩固所学知识，并尝试将它们结合起来解决更复杂的问题。

第4章 STL的基本使用学习指南

4.1 STL 中的数据结构

目标：熟悉STL中各种数据结构的特点和使用场景。

内容：

- 向量（vector）的定义和操作方法。
- 栈（stack）的定义和操作方法。
- 队列（queue）的定义和操作方法。
- map的定义和操作方法。
- 堆优先队列（priority_queue）的定义和操作方法。
- 集合（set）的定义和操作方法。
- 多重集合（multiset）的定义和操作方法。
- 双端队列（deque）的定义和操作方法。
- string的定义和操作方法。
- pair的定义和操作方法。
- bitset的定义和操作方法。

学习方法：

- 查阅STL文档，了解每种数据结构的接口和成员函数。
- 通过编程练习，熟悉每种数据结构的基本操作和性能特点。

4.2 STL 中的算法

目标：熟悉STL中提供的常用算法及其使用方法。

内容：

- sort()函数的使用和自定义排序规则。
- lower_bound()和upper_bound()函数的使用和应用场景。
- reverse()函数的使用和效果展示。
- swap()函数的使用和注意事项。
- next_permutation()和prev_permutation()函数的使用和效果展示。

学习方法：

- 查阅STL算法的文档，了解每个算法的功能和参数要求。
- 通过编程练习，掌握每个算法的使用方法和效果展示。

第5章 搜索学习指南

5.1 深度优先搜索（回溯法）

目标：掌握深度优先搜索和回溯法的基本思想和应用。

内容：

- 子集树的构建和应用。
- 排列树的生成和使用场景。
- FloodFill算法的原理与实现。
- 通过例题讲解加深对深度优先搜索的理解。

学习方法：

- 理解并实现子集树和排列树的递归构建过程。
- 实践FloodFill算法，如图像填充、迷宫求解等。
- 通过解决经典的深度优先搜索问题来巩固所学知识。

5.2 广度优先搜索

目标：掌握广度优先搜索的基本思想和应用场景。

内容：

- 等权的最短路径问题的解决方法。
- 最少操作次数问题的解决策略。

学习方法：

- 理解并实现广度优先搜索算法，如队列的应用。
- 通过解决实际问题，如最短路径、最小生成树等，来加深对广度优先搜索的理解。

5.3 搜索的优化方法

目标：了解并应用搜索算法的优化方法。

内容：

- 剪枝技术的原理和应用。
- 记忆化搜索的方法和优势。
- 通过例题讲解加深对搜索优化方法的理解。

学习方法：

- 学习并实践剪枝技术，如Alpha-Beta剪枝、分支限界等。

- 掌握记忆化搜索的思想，并通过编程实现来提高算法效率。
- 通过解决复杂的搜索问题来验证优化方法的有效性。

第 6 章 动态规划学习指南

6.1 动态规划基础

目标：理解动态规划的基本概念和原理，掌握其基本步骤。

内容：

- 状态的定义和表示方法。
- 状态转移方程的建立和应用。
- 注意边界条件的处理。
- 做题的基本步骤和方法。

学习方法：

- 通过实例学习如何定义状态和建立状态转移方程。
- 理解并应用动态规划的基本步骤，如自底向上、自顶向下等。

6.2 背包 DP

目标：掌握背包动态规划问题的解法，包括01背包、完全背包和多重背包。

内容：

- 01背包问题的解决方法。
- 完全背包问题的解决方法。
- 多重背包问题的解决方法。
- 通过例题讲解加深对背包动态规划的理解。

学习方法：

- 分别学习和实践01背包、完全背包和多重背包的动态规划解法。
- 通过解决实际问题来巩固所学知识，如资源分配、货物装载等。

6.3 区间 DP

目标：掌握区间动态规划问题的解法。

内容：

- 石子合并问题的解决方法。
- 其他区间动态规划问题的解题思路。

学习方法：

- 通过石子合并问题学习区间动态规划的解法。
- 尝试将区间动态规划应用于其他类似问题，如区间调度、区间覆盖等。

6.4 存在性 DP

目标：理解存在性动态规划问题的解法。

内容：

- 什么是存在性动态规划问题。
- 通过例题讲解加深对存在性动态规划的理解。

学习方法：

- 学习并理解存在性动态规划问题的特点和解法。
- 通过解决实际问题来巩固所学知识，如子序列、子数组等问题。

6.5 状压 DP

目标：掌握状态压缩动态规划问题的解法。

内容：

- 状态压缩的方法和技巧。
- 通过例题讲解加深对状态压缩动态规划的理解。

学习方法：

- 学习并理解状态压缩的思想和方法。
- 通过编程实践来掌握状态压缩的技巧，如位运算、哈希表等。

6.6 期望 DP

目标：理解期望动态规划问题的解法。

内容：

- 期望的性质和转移方式。
- 通过例题讲解加深对期望动态规划的理解。

学习方法：

- 学习并理解期望动态规划的特点和解法。
- 通过解决实际问题来巩固所学知识，如随机游走、赌博游戏等。

6.7 树形 DP

目标：掌握树形动态规划问题的解法。

内容：

- 树形动态规划的介绍和基本原理。
- 自下而上树形动态规划的解法。
- 换根动态规划的解法。
- 通过例题讲解加深对树形动态规划的理解。

学习方法：

- 学习并理解树形动态规划的特点和解法。
- 通过编程实践来掌握树形动态规划的技巧，如树的遍历、动态规划的状态转移等。

第 7 章 图论学习指南

7.1 图的存储方法

目标：掌握不同图的存储方法及其适用场景。

内容：

- 邻接矩阵的表示与应用。
- 邻接表的构建与使用。
- 链式前向星的实现与优点。

学习方法：

- 理解并实践每种存储方法，通过代码实现加深印象。
- 比较不同存储方法的优缺点，选择适合特定问题的存储方式。

7.2 图上问题

目标：掌握常见的图算法及其应用。

内容：

- 图的分类和性质，如连通性、平面性等。
- 图的遍历方法，如深度优先搜索（DFS）和广度优先搜索（BFS）。
- Dijkstra最短路径算法及其优化。
- Bellman-Ford最短路径算法及其用途。
- Johnson最短路径算法的特点。

- Floyd最短路径算法的原理。
- 匈牙利算法在二分图匹配中的应用。
- Tarjan算法在强连通分量中的应用。
- DAG-DP动态规划在有向无环图中的应用。

学习方法：

- 逐一学习和实践上述算法，通过解决实际问题来巩固所学知识。
- 分析并比较不同算法的时间复杂度和空间复杂度，选择最适合特定问题的算法。

7.3 树上问题

目标：掌握树结构相关问题的解决方法。

内容：

- 树的基本概念和性质。
- 最小生成树的求解方法，如Prim和Kruskal算法。
- 倍增LCA（最近公共祖先）的求解技巧。
- 例题讲解，涵盖树上问题的多种解法。

学习方法：

- 通过实例学习树的性质和相关算法。
- 解决实际问题，如最短路径、直径等，来加深对树结构和算法的理解。

第 8 章 进阶数据结构学习指南

8.1 单调栈

目标：掌握单调栈的数据结构及其应用。

内容：

- 单调栈的介绍和基本原理。
- 例题讲解，展示单调栈在解决问题中的作用。

学习方法：

- 理解单调栈的工作原理，通过编程实践来熟悉其操作。
- 尝试将单调栈应用于不同的问题场景，如括号匹配、滑动窗口最大值等。

8.2 单调队列

目标：掌握单调队列的数据结构及其应用。

内容：

- 单调队列的介绍和基本原理。
- 例题讲解，展示单调队列在解决问题中的作用。

学习方法：

- 理解单调队列的工作原理，通过编程实践来熟悉其操作。
- 尝试将单调队列应用于不同的问题场景，如双端滑动窗口最值等。

8.3 ST 表

目标：掌握ST表的数据结构及其应用。

内容：

- ST表的介绍和基本原理。
- 例题讲解，展示ST表在解决问题中的作用。

学习方法：

- 理解ST表的工作原理，通过编程实践来熟悉其操作。
- 尝试将ST表应用于不同的问题场景，如区间查询等。

8.4 树状数组

目标：掌握树状数组的数据结构及其应用。

内容：

- 单点修改型树状数组的实现与应用。
- 区间修改型树状数组的实现与应用。
- 例题讲解，展示树状数组在解决问题中的作用。

学习方法：

- 理解树状数组的工作原理，通过编程实践来熟悉其操作。
- 尝试将树状数组应用于不同的问题场景，如区间加法、区间最大值等。

8.5 线段树

目标：掌握线段树的数据结构及其应用。

内容：

- 线段树区间加法的实现与应用。
- 线段树的区间乘法、加法和赋值操作。
- 例题讲解，展示线段树在解决问题中的作用。

学习方法：

- 理解线段树的工作原理，通过编程实践来熟悉其操作。
- 尝试将线段树应用于不同的问题场景，如区间求和、区间最大值等。

8.6 并查集

目标：掌握并查集的数据结构及其应用。

内容：

- 朴素并查集的实现与应用。
- 并查集的路径压缩优化。
- 并查集的启发式合并优化。
- 可撤销并查集的实现与应用。
- 例题讲解，展示并查集在解决问题中的作用。

学习方法：

- 理解并查集的工作原理，通过编程实践来熟悉其操作。
- 尝试将并查集应用于不同的问题场景，如判断连通性、查找集合的代表元素等。

8.7 链表

目标：掌握链表的数据结构及其应用。

内容：

- 数组实现双向链表的方法。
- 例题讲解，展示链表在解决问题中的作用。

学习方法：

- 理解链表的工作原理，通过编程实践来熟悉其操作。
- 尝试将链表应用于不同的问题场景，如约瑟夫环、多项式相加等。

第 9 章 字符串学习指南

9.1 字符串匹配

目标：掌握不同字符串匹配算法的原理和应用。

内容：

- 朴素的字符串匹配算法及其实现。
- KMP算法的原理和实现。
- 进制哈希的原理和应用。
- 通过例题讲解加深对字符串匹配算法的理解。

学习方法：

- 理解并实践每种字符串匹配算法，通过编程实现来加深印象。
- 比较不同算法的时间复杂度和空间复杂度，选择适合特定问题的算法。

9.2 回文串

目标：掌握回文串的定义、性质及算法。

内容：

- 回文串的基本概念和性质。
- Manacher算法的原理和实现。
- 通过例题讲解加深对回文串算法的理解。

学习方法：

- 理解Manacher算法的工作原理，通过编程实践来熟悉其操作。
- 尝试将Manacher算法应用于不同的问题场景，如最长回文子串等。

9.3 Trie 树（字典树）

目标：掌握Trie树的数据结构及其应用。

内容：

- Trie树的基本概念和特点。
- 字符Trie树的构建和应用。
- 01Trie树的构建和应用。
- 通过例题讲解加深对Trie树的理解。

学习方法：

- 理解Trie树的工作原理，通过编程实践来熟悉其操作。
- 尝试将Trie树应用于不同的问题场景，如单词查找、自动补全等。

第 10 章 数论学习指南

10.1 数论基础

目标：掌握数论的基本概念和性质。

内容：

- 数论的讨论范围和基本概念。
- 整数除法的性质和定理。
- 取模运算的性质和定理。

学习方法：

- 理解并记忆数论的基本概念和性质，通过例题加深印象。
- 尝试解决简单的数论问题，如整除性判断、同余方程等。

10.2 唯一分解定理和约数定理

目标：掌握唯一分解定理和约数定理的原理和应用。

内容：

- 唯一分解定理的内容和证明。
- 约数定理的内容和应用。
- 因数分解和质因数分解的方法。
- 通过例题讲解加深对定理的理解。

学习方法：

- 理解并证明唯一分解定理和约数定理，通过编程实践来加深印象。
- 尝试将定理应用于不同的问题场景，如求最大公约数、最小公倍数等。

10.3 最大公约数和最小公倍数

目标：掌握最大公约数和最小公倍数的计算方法。

内容：

- 辗转相除法的原理和应用。
- 最大公约数和最小公倍数在唯一分解中的性质。

学习方法：

- 理解并实践辗转相除法，通过编程实现来加深印象。
- 尝试将算法应用于不同的问题场景，如分数加减、比例问题等。

10.4 拓展欧几里得

目标：掌握拓展欧几里得算法的原理和应用。

内容：

- 裴蜀定理的内容和证明。
- 拓展欧几里得算法的原理和应用。

学习方法：

- 理解并证明裴蜀定理，通过编程实践来加深印象。
- 尝试将算法应用于不同的问题场景，如求解线性不定方程等。

10.5 快速幂

目标：掌握快速幂的原理和模板。

内容：

- 为什么要用快速幂及其应用场景。
- 快速幂的原理和模板实现。

学习方法：

- 理解快速幂的原理，通过编程实践来熟悉其操作。
- 尝试将快速幂应用于不同的问题场景，如大数取模、矩阵快速幂等。

10.6 乘法逆元

目标：掌握乘法逆元的表示方法和求解方法。

内容：

- 乘法逆元如何表示除法及其应用场景。
- 费马小定理求逆元的原理和方法。

学习方法：

- 理解并实践费马小定理求逆元的方法，通过编程实现来加深印象。
- 尝试将算法应用于不同的问题场景，如RSA加密、模逆元求解等。

10.7 组合计数

目标：掌握组合计数的基本方法和定理。

内容：

- 分类加法和分步乘法的应用。
- 组合数的定义和性质。
- 普通型生成函数的原理和应用。
- Lucas定理的内容和应用。

学习方法：

- 理解并记忆组合计数的基本方法和定理，通过例题加深印象。
- 尝试解决实际的组合计数问题，如排列组合、概率问题等。

10.8 关于质数的判断

目标：掌握质数判断的方法和算法。

内容：

- 单点质数判断（试除法）的原理和应用。
- 埃氏筛法的原理和应用。
- 欧拉筛法的原理和应用。

学习方法：

- 理解并实践各种质数判断算法，通过编程实现来加深印象。
- 尝试将算法应用于不同的问题场景，如素数筛选、质因数分解等。

10.9 欧拉函数

目标：掌握欧拉函数的定义、性质及算法。

内容：

- 单点欧拉函数的定义和计算方法。
- 筛法求欧拉函数的原理和应用。
- 欧拉定理的内容和应用。
- 欧拉降幂的原理和应用。

学习方法：

- 理解并记忆欧拉函数的定义、性质及算法，通过例题加深印象。
- 尝试解决实际欧拉函数问题，如欧拉路径、图的遍历等。

10.10 异或线性基

目标：掌握异或线性基的原理和性质。

内容：

- 异或线性基的原理和性质。

学习方法：

- 理解异或线性基的原理和性质，通过编程实践来加深印象。
- 尝试将异或线性基应用于不同的问题场景，如位运算优化、数据压缩等。

第 11 章 博弈论学习指南

11.1 基础博弈类型

目标：掌握基础博弈类型的定义、性质及解法。

内容：

- Bash博弈的定义和策略。
- Nim博弈的定义和策略。
- 例题讲解，加深对Bash博弈和Nim博弈的理解。

学习方法：

- 理解并记忆Bash博弈和Nim博弈的定义和性质。
- 通过编程实践解决例题，加深对博弈策略的理解。

11.2 SG 函数

目标：掌握SG函数的定义、性质及应用。

内容：

- mex运算的定义和计算方法。
- SG函数的定义和性质。
- 子游戏的合并技巧。
- SG函数打表方法。
- 例题讲解，加深对SG函数的理解。

学习方法：

- 理解并记忆SG函数的定义和性质。

- 通过编程实践解决例题，加深对SG函数的应用能力。

11.3 反 Nim 博弈

目标：掌握反Nim博弈的结论、证明及应用。

内容：

- 反Nim博弈的结论。
- 结论的证明。
- 例题讲解，加深对反Nim博弈的理解。

学习方法：

- 理解并记忆反Nim博弈的结论和证明。
- 通过编程实践解决例题，加深对反Nim博弈的应用能力。

11.4 博弈杂题选讲

目标：掌握博弈论中的杂题类型及解法。

内容：博弈论中的杂题选讲，包括各种不同类型的博弈问题。

学习方法：

- 理解并记忆博弈论中的杂题类型和解法。
- 通过编程实践解决例题，加深对博弈论杂题的理解和应用能力。

第 12 章 高级算法策略与技巧学习指南

12.1 构造

目标：掌握构造问题的常见思维和解题策略。

内容：

- 构造问题的常见思维。
- 例题讲解，加深对构造问题的理解。

学习方法：

- 理解并记忆构造问题的常见思维和解题策略。
- 通过编程实践解决例题，加深对构造问题的理解和应用能力。

12.2 分块思想

目标：掌握分块思想的优化方法和应用场景。

内容：

- 根号分块优化方法。
- 整除分块方法。

学习方法：

- 理解并记忆分块思想的优化方法和应用场景。
- 通过编程实践解决例题，加深对分块思想的理解和应用能力。

12.3 离散化

目标：掌握离散化的基本思想和应用场景。

内容：离散化的基本思想和应用场景介绍。

学习方法：

- 理解并记忆离散化的基本思想和应用场景。
- 通过编程实践解决例题，加深对离散化的理解和应用能力。

12.4 离线思想

目标：掌握离线思想的基本思想和应用场景。

内容：离线思想的基本思想和应用场景介绍。

学习方法：

- 理解并记忆离线思想的基本思想和应用场景。
- 通过编程实践解决例题，加深对离线思想的理解和应用能力。

12.5 莫队算法

目标：掌握莫队算法的原理和应用。

内容：

- 莫队算法的介绍。
- 例题讲解，加深对莫队算法的理解。

学习方法：

- 理解并记忆莫队算法的原理和应用。
- 通过编程实践解决例题，加深对莫队算法的理解和应用能力。

12.6 CDQ 分治

目标：掌握CDQ分治算法的原理和应用。

内容:

- 点对/区间相关问题的CDQ分治算法。
- 三维偏序问题的CDQ分治算法。

学习方法:

- 理解并记忆CDQ分治算法的原理和应用。
- 通过编程实践解决例题,加深对CDQ分治算法的理解和应用能力。

12.7 本章小结

目标: 回顾并总结本章所学内容。

内容: 本章内容的小结和复习。

学习方法:

- 回顾并总结本章所学内容,加深理解和记忆。
- 通过解决本章例题,检验学习效果并巩固所学知识。

第 13 章 真题选讲学习指南

13.1 XCPC 往年真题选讲

目标: 熟悉 XCPC 往年真题的解题思路和方法。

内容: XCPC 往年真题的选讲和解析。

学习方法:

- 理解并记忆XCPC往年真题的解题思路和方法。
- 通过编程实践解决真题,提高解题能力和水平。

13.2 NOI/NOIP 往年真题选讲

目标: 熟悉NOI/NOIP往年真题的解题思路和方法。

内容: NOI/NOIP往年真题的选讲和解析。

学习方法:

- 理解并记忆NOI/NOIP往年真题的解题思路和方法。
- 通过编程实践解决真题,提高解题能力和水平。

13.3 蓝桥杯往年真题选讲

目标: 熟悉蓝桥杯往年真题的解题思路和方法。

内容：蓝桥杯往年真题的选讲和解析。

学习方法：

- 理解并记忆蓝桥杯往年真题的解题思路和方法。
- 通过编程实践解决真题，提高解题能力和水平。

13.4 天梯赛往年真题选讲

目标：熟悉天梯赛往年真题的解题思路和方法。

内容：天梯赛往年真题的选讲和解析。

学习方法：

- 理解并记忆天梯赛往年真题的解题思路和方法。
- 通过编程实践解决真题，提高解题能力和水平。