



National Olympiad in Informatics

全国青少年信息学奥林匹克系列 竞赛大纲 (2025年修订版)

CCF NOI 科学委员会全体审定



中國計算機學會
CHINA COMPUTER FEDERATION

全国青少年信息学奥林匹克竞赛 (CCF NOI)

Email: noi@ccf.org.cn

网址: <http://www.noi.cn>



中國計算機學會
CHINA COMPUTER FEDERATION



National Olympiad in Informatics

全国青少年信息学奥林匹克系列

竞赛大纲 (2025年修订版)

CCF NOI 科学委员会全体审定



中國計算機學會
CHINA COMPUTER FEDERATION

>>>> 序 <<<<<

全国青少年信息学奥林匹克竞赛（NOI）于1984年创办，至今已经四十余载。NOI从无到有，通过持续发展革新，已经从单一的小规模的编程赛事发展为形式丰富多样、参加人数众多的系列活动，最近每年直接参加活动的人数已逾20万，培养了一批在计算机科学方面有兴趣和特长的学生，也源源不断为大学输送了优质生源。这些学生在高校接受了计算机专业的学术训练，毕业后成为学术界、工业界的领军人物和中坚力量。

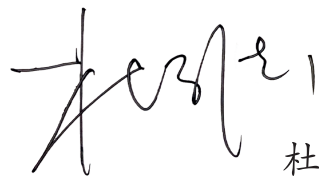
在过去两年内，以GPT、DeepSeek为代表的大语言模型，在人工智能领域取得了令人振奋的突破，在程序和算法设计方面也取得了良好的进展。在此形势下，NOI需要优化考查方向与知识体系，以凸显人类计算思维在算法设计中独有的创造性。同时，NOI仍然面临着优质师资短缺、区域发展不平衡的双重挑战，为此需要继续加强知识体系的合理性和规范性，以缩小地区、学校之间的“信息差”，促进师资的培养和活动的普及。

CCF NOI科学委员会与全国数十位NOI资深指导教师共同努力，于2021年制定并首次发布了NOI大纲，并在2023年做了第一次修订。大纲对竞赛组织、

教师教学和选手学习都发挥了很好的引导作用。为了适应计算机科学技术的发展，大纲工作组在2023年修订版的基础上，通过对国内外数十个主流信息学比赛的调研，以及对全国NOI师生在内各方意见的征询，历时两年完成了本次大纲修订工作。

我相信，NOI大纲将会持续对NOI以及计算机科学普及等活动发挥积极的引导作用。同时，NOI科学委员会也秉承CCF一向坚持的开放原则，诚挚希望广大师生能够在教学和竞赛的过程中，将发现的问题及时反馈给大纲工作组或NOI科学委员会（noi@ccf.org.cn），以便我们更好地完善这部大纲，共同推动NOI活动的发展。

我代表 CCF 感谢 NOI 科学委员会对该项工作持续的关注和贡献。如果没有诸位委员的付出，这份大纲的制定和更新就无从实现。特别感谢科学委员会副主席赵启阳博士，他牵头并组织众多专家完成了这次修订工作。同时，还有来自中学的诸位资深教师以及付出辛勤劳动的人员，我对他们的贡献也致以深深的谢意。



杜子德

全国青少年信息学奥林匹克竞赛（NOI）主席

2025 年 4 月

全国青少年信息学奥林匹克系列竞赛大纲

2025版修订：赵启阳(召集人)、叶金毅、胡伟栋、
金靖、李建、谢秋锋、汪星明、
李曙、叶国平

2025版调研：谢秋锋、金靖、李曙、艾庆兴、
陈睿、陈许旻、陈奕哲、董华星、
黎凡、潘玉斌、王晓鹏、肖然、
杨静、余岳、张超、周鹏、
诸一行

2023版修订：赵启阳(召集人)、叶金毅、胡伟栋、
金靖、李建、叶国平、汪星明、
谢秋锋、李曙

2021版起草：朱全民、江涛、宋新波、汪星明、
胡伟栋、赵启阳

2021版审阅：韩文弢、陈合力、徐先友、曹利国、
叶金毅、史习艺、滕伟、邱桂香、
李曙、叶国平、王晓光、蔺洋、
符水波、金靖、贾志勇、杨森林、
曾艺卿、马建辉、吴旭日、蒋帅、
缪和平、洪嘉毅、韩冰、
彭决(特邀)、钟诚(特邀)

日期：2025-4-15

>>>> 目录 <<<<

全国青少年信息学奥林匹克系列竞赛大纲

一、简 介 3

1.1 目的	3
1.2 原则	4
1.3 建议	7
1.4 修订	8
1.5 致谢	8

二、内 容 9

2.1 入门级	9
2.2 提高级	19
2.3 NOI级	26

>>>> 一、简介 <<<<

1.1 目的

本大纲的制定目的在于：

- (1) 为全国青少年信息学奥林匹克（National Olympiad in Informatics, NOI）系列竞赛以及中国计算机学会（China Computer Federation, CCF）主办的其他有关活动的题目命制提供依据；
- (2) 为NOI指导教师的教学提供方向和指导；
- (3) 为参加NOI系列竞赛、CCF主办的其他有关活动的学生和信息学爱好者的学习提供范围；
- (4) 为各省市开展和组织NOI省选等活动提供参照。



1.2 原则

1.2.1 等级化原则

按照目前NOI系列活动开展现状，以及将来可能的发展，大纲将各知识点分成入门级、提高级和NOI级。高级别自动包含低级别知识点。各级别与NOI系列活动以及CCF主办的其他活动的对应关系如下：

(1) 入门级：CCF非专业级软件能力认证入门组（Certified Software Professional Junior, CSP-J）；

(2) 提高级：全国青少年信息学奥林匹克联赛（National Olympiad in Informatics in Provinces, NOIP）、CCF非专业级软件能力认证提高组（Certified Software Professional Senior, CSP-S）；

(3) NOI级：全国青少年信息学奥林匹克竞赛（NOI）及以上，包括国际信息学奥林匹克（International Olympiad in Informatics, IOI）中国队选拔（China Team Selection, CTS）、NOI冬令营、国家集训队集训等。

除将所有知识点划分为上述级别以外，还对每个知识点标定了学习难度系数（范围为1~10）。考虑到相邻级别知识点的学习难度可能存在交错，因此将入门级知识点的难度系数范围设置为1~5，（除入门级知识点外的）提高级知识点的难度系数范围设置为5~8，

（除入门级、提高级知识点外的）NOI级知识点的难度系数范围设置为7~10。

各知识点难度系数以【X】的格式列在知识点之前。

1.2.2 差异化原则

为促进信息学和NOI活动的普及，大纲应较详尽地规定中低级别知识点的范围，以尽可能清晰地划定相应级别的知识范围，有效地指导入门学生的学习及相关的教学活动；为保证和促进我国选手在IOI竞赛中的竞争力，大纲应避免过于严格地限制命题的思路，须为NOI等高水平竞赛的题目命制留有充分的开放性，因此不宜过于细致地规定高级别知识点的范围。为此，大纲在制定中将采取“上粗下细”的指导思想：知识级别越低，其内容规定得越细；知识级别越高，其内容规定得越粗。

1.2.3 统一性原则

为保证大纲的简明性和系统性，高级别比赛的知识范围将自动地包含低级别比赛的所有知识点。同时，对每个级别按照竞赛环境（Linux和Windows）、程序设计语言（C++）、数据结构、算法以及数学等进行了分类。对每个大类又按照知识点的属性继续划



分为若干小类；某些知识点可能与多个类别均有紧密或松散联系，本大纲均按其主要属性划定其类别，以避免同一知识点在多个类别中的重复出现。



1.3 建议

建议在各级别竞赛题目的命制中，

- (1) 各级别竞赛或活动的考察范围不超过对应的大纲级别，其中难度系数为10的知识点仅用于CTS；
- (2) 避免对算法复杂度的常系数的考察；
- (3) 部分单个知识点可能对应不同层次、不同性能的多个数据结构或算法。考察内容应以常见的、经典的内容为主，避免虽具有微弱性能优势（例如算法复杂度的细微改进）但较为冷僻或过新的数据结构和算法。





1.4 修 订

(1) 大纲将根据 NOI 活动的发展而定期进行维护和修订，修订周期为两年；

(2) 本轮大纲维护小组成员为：赵启阳（召集人）、叶金毅、胡伟栋、金靖、李建、谢秋锋、汪星明、李曙、叶国平。欢迎将修订意见反馈给以上人员。

1.5 致 谢

在本轮大纲的修订过程中，杨耀良、崔浩以及多位未具名的师生均提出了各种宝贵意见，在此表示感谢。



>>>> 二、内 容 <<<<

2.1 入门级

2.1.1 基础知识与编程环境

1. **【1】** 计算机的基本构成（CPU、内存、I/O设备等）
2. **【1】** Windows、Linux等操作系统的基本概念及其常见操作
3. **【1】** 计算机网络和Internet的基本概念
4. **【1】** 计算机的历史和常见用途
5. **【1】** NOI以及相关活动的历史
6. **【1】** NOI以及相关活动的规则
7. **【1】** 位、字节与字
8. **【1】** 程序设计语言以及程序编译和运行的基本概念
9. **【1】** 使用图形界面新建、复制、删除、移动文件或目录
10. **【1】** 使用 Windows 系统下的集成开发环境（例如Dev-C++等）
11. **【1】** 使用Linux系统下的集成开发环境（例

【x】 表示难度系数

如Code::Blocks等)

12. 【1】 常用编译命令g++的基本使用

2.1.2 C++程序设计¹

1. 程序基本概念

- 【1】 标识符、关键字、常量、变量、字符串、表达式的概念
- 【1】 常量与变量的命名、定义及作用
- 【2】 头文件与名字空间的概念
- 【2】 编辑、编译、解释、调试的概念

2. 基本数据类型

- 【1】 整数型: int、long long
- 【1】 实数型: float、double
- 【1】 字符型: char
- 【1】 布尔型: bool

3. 程序基本语句

- 【2】 cin语句、scanf语句、cout语句、printf语句、赋值语句、复合语句

¹ C++ 以外的其他高级程序设计语言可参照本部分内容。

- 【2】 if语句、switch语句、多层条件语句
- 【2】 for语句、while语句、do while语句
- 【3】 多层循环语句

4. 基本运算

- 【1】 算术运算: 加、减、乘、除、整除、求余
- 【1】 关系运算: 大于、大于等于、小于、小于等于、等于、不等于
- 【1】 逻辑运算: 与 (&&)、或 (||)、非 (!)
- 【1】 变量自增与自减运算
- 【1】 三目运算
- 【2】 位运算: 与 (&)、或 (|)、非 (~)、异或 (^)、左移 (<<)、右移 (>>)

5. 数学库常用函数

- 【3】 绝对值函数、四舍五入函数、下取整函数、上取整函数、平方根函数、常用三角函数、对数函数、指数函数

6. 结构化程序设计

- 【1】 顺序结构、分支结构和循环结构
- 【2】 自顶向下、逐步求精的模块化程序设计

- 【2】 流程图的概念及流程图描述

7. 数组

- 【1】 数组与数组下标
- 【1】 数组的读入与输出
- 【3】 二维数组与多维数组

8. 字符串的处理

- 【2】 字符数组与相关函数
- 【2】 string类与相关函数

9. 函数与递归

- 【2】 函数定义与调用、形参与实参
- 【3】 传值参数与传引用参数
- 【2】 常量与变量的作用范围
- 【2】 递归函数

10. 结构体与联合体

- 【3】 结构体
- 【3】 联合体

11. 指针与引用

- 【4】 指针
- 【4】 基于指针的数组访问
- 【4】 字符指针
- 【4】 指向结构体的指针
- 【5】 引用

12. 文件及基本读写

- 【2】 文件的基本概念、文本文件的基本操作
- 【2】 文本文件类型与二进制文件类型
- 【2】 文件重定向、文件读写等操作

13. STL模板

- 【3】 常用函数与算法模板: min、max、swap、sort
- 【4】 栈 (stack)、队列 (queue)、链表 (list)、向量 (vector) 等容器

2.1.3 数据结构

1. 线性结构

- 【3】 链表: 单链表、双向链表、循环链表
- 【3】 栈
- 【3】 队列

2. 简单树

- 【3】 树的定义与相关概念
- 【4】 树的表示与存储
- 【3】 二叉树的定义与基本性质
- 【4】 二叉树的表示与存储
- 【4】 二叉树的遍历：前序、中序、后序

3. 特殊树

- 【4】 完全二叉树的定义与基本性质
- 【4】 完全二叉树的数组表示法
- 【4】 哈夫曼树的定义和构造、哈夫曼编码
- 【4】 二叉搜索树的定义和构造

4. 简单图

- 【3】 图的定义与相关概念
- 【4】 图的表示与存储：邻接矩阵
- 【4】 图的表示与存储：邻接表

2.1.4 算法

1. 算法概念与描述

- 【1】 算法概念

- 【2】 算法描述：自然语言描述、流程图描述、伪代码描述

2. 入门算法

- 【1】 枚举法
- 【1】 模拟法

3. 基础算法

- 【3】 贪心法
- 【3】 递推法
- 【4】 递归法
- 【4】 二分法
- 【4】 倍增法

4. 算法策略

- 【3】 前缀和
- 【4】 差分

5. 数值处理算法

- 【4】 高精度的加法
- 【4】 高精度的减法
- 【4】 高精度的乘法

- 【4】 高精度整数除以单精度整数的商和余数

6. 排序算法

- 【3】 排序的基本概念
- 【3】 冒泡排序
- 【3】 选择排序
- 【3】 插入排序
- 【3】 计数排序

7. 搜索算法

- 【5】 深度优先搜索
- 【5】 广度优先搜索

8. 图论算法

- 【4】 深度优先遍历
- 【4】 广度优先遍历
- 【5】 泛洪算法 (Flood Fill)

9. 动态规划

- 【4】 动态规划的基本思路
- 【4】 简单一维动态规划
- 【5】 简单背包类型动态规划

- 【5】 简单区间类型动态规划

2.1.5 数学与其他

1. 数及其运算

- 【1】 自然数、整数、有理数、实数及其算术运算 (加、减、乘、除)
- 【1】 进制与进制转换：二进制、八进制、十进制、十六进制

2. 初等数学

- 【1】 代数 (初中部分)
- 【1】 几何 (初中部分)

3. 初等数论

- 【3】 整除、因数、倍数、指数、质 (素) 数、合数
- 【3】 取整
- 【3】 模运算与取余
- 【3】 整数唯一分解定理
- 【3】 辗转相除法 (欧几里得算法)
- 【4】 素数筛法：埃氏筛法与线性筛法



4.离散与组合数学

- 【2】 集合
- 【2】 加法原理
- 【2】 乘法原理
- 【4】 排列
- 【4】 组合
- 【4】 杨辉三角

5.其他

- 【2】 ASCII码



2.2 提高级

2.2.1 基础知识与编程环境

1. 【5】 Linux系统终端中常用的文件与目录操作命令
2. 【5】 Linux系统下常见文本编辑工具的使用
3. 【5】 常用编译命令g++与相关编译选项
4. 【5】 在Linux系统终端中运行程序，使用time命令查看程序用时
5. 【5】 调试工具GDB的使用

2.2.2 C++程序设计²

1.类 (class)

- 【6】 类的概念及简单应用
- 【6】 成员函数和运算符重载

2.STL模板

- 【5】 容器 (container) 和迭代器 (iterator)
- 【5】 对 (pair)、元组 (tuple)

²C++ 以外的其他高级程序设计语言可参照本部分内容。

- 【5】 集合 (set)、多重集合 (multiset)
- 【5】 双端队列 (deque)、优先队列 (priority_queue)
- 【5】 映射 (map)、多重映射 (multimap)
- 【5】 位集合 (bitset)
- 【5】 算法模板库中的常用函数

2.2.3 数据结构

1. 线性结构

- 【5】 双端栈
- 【5】 双端队列
- 【5】 单调队列
- 【6】 优先队列
- 【6】 ST表 (Sparse Table)

2. 集合与森林

- 【6】 并查集
- 【6】 树的孩子兄弟表示法

3. 特殊树

- 【6】 二叉堆

- 【6】 树状数组
- 【6】 线段树
- 【6】 字典树 (Trie)
- 【7】 笛卡尔树
- 【8】 平衡树: AVL、Treap、Splay等

4. 常见图

- 【5】 稀疏图
- 【6】 偶图 (二分图)
- 【6】 欧拉图
- 【6】 有向无环图
- 【7】 连通图与强连通图
- 【7】 双连通图

5. 哈希表

- 【5】 数值哈希函数构造
- 【6】 字符串哈希函数构造
- 【6】 哈希冲突的常用处理方法

2.2.4 算法

1. 复杂度分析

- 【6】 时间复杂度分析

- 【6】 空间复杂度分析

2. 算法策略

- 【6】 离散化
- 【7】 扫描线

3. 基础算法

- 【6】 分治算法

4. 排序算法

- 【5】 归并排序
- 【5】 快速排序
- 【6】 堆排序
- 【5】 桶排序
- 【6】 基数排序

5. 字符串算法

- 【6】 字符串匹配：KMP算法
- 【7】 Manacher算法

6. 搜索算法

- 【6】 搜索的剪枝优化

- 【6】 记忆化搜索
- 【7】 启发式搜索
- 【7】 双向广度优先搜索
- 【7】 迭代加深搜索

7. 图论算法

- 【6】 最小生成树：Prim和Kruskal等算法
- 【6】 单源最短路：Bellman-Ford、Dijkstra、SPFA等算法
- 【7】 单源次短路
- 【6】 Floyd-Warshall算法
- 【6】 有向无环图的拓扑排序
- 【6】 欧拉道路和欧拉回路
- 【6】 二分图的判定
- 【7】 强连通分量
- 【7】 割点、割边
- 【6】 树的重心、直径、DFS序与欧拉序
- 【6】 树上差分、子树和与倍增
- 【6】 最近公共祖先

8. 动态规划

- 【6】 多维动态规划

- 【6】 树型动态规划
- 【7】 状态压缩动态规划
- 【8】 动态规划的常用优化

2.2.5 数学与其他

1.初等数学

- 【5】 代数（高中部分）
- 【6】 几何（高中部分）

2.初等数论

- 【5】 同余式
- 【7】 欧拉定理和欧拉函数
- 【7】 费马小定理
- 【7】 威尔逊定理
- 【7】 裴蜀定理
- 【7】 模运算意义下的逆元
- 【7】 扩展欧几里得算法
- 【7】 中国剩余定理

3.离散与组合数学

- 【6】 多重集合

- 【6】 等价关系与等价类
- 【6】 多重集上的排列
- 【6】 多重集上的组合
- 【6】 错排列、圆排列
- 【6】 鸽巢原理
- 【6】 二项式定理
- 【7】 容斥原理
- 【7】 卡特兰（Catalan）数

4.线性代数

- 【5】 向量与矩阵的概念
- 【6】 向量的运算
- 【6】 矩阵的初等变换
- 【6】 矩阵的运算：加法、减法、乘法与转置
- 【6】 特殊矩阵的概念：单位阵、三角阵、对称阵和稀疏矩阵
- 【7】 高斯消元法

2.3 NOI级

2.3.1 C++程序设计³

1. 【8】面向对象的程序设计思想（OOP）

2.3.2 数据结构

1.线性结构

- 【8】块状链表

2.复杂树

- 【8】树链剖分
- 【10】动态树：LCT
- 【9】树套树
- 【9】k-d树
- 【8】虚树

3.可合并堆

- 【8】左偏树
- 【10】二项堆

³ C++ 以外的其他高级程序设计语言可参照本部分内容。

4.可持久化数据结构

- 【8】可持久化线段树
- 【9】其他可持久化数据结构

2.3.3 算法

1.算法策略

- 【8】分块
- 【8】离线处理思想
- 【9】复杂分治思想
- 【9】平衡规划思想
- 【9】构造思想

2.字符串算法

- 【8】扩展KMP算法
- 【8】有穷自动机的概念
- 【8】AC自动机
- 【8】后缀数组
- 【9】后缀树
- 【10】后缀自动机

3.图论算法

- 【8】 基环树
- 【10】 最小树形图
- 【8】 2-SAT
- 【8】 网络流
- 【10】 图的支配集、独立集与覆盖集
- 【8】 匈牙利算法
- 【10】 KM算法
- 【10】 一般图的匹配

4.动态规划

- 【9】 复杂动态规划模型的构建
- 【9】 复杂动态规划模型的优化

2.3.4 数学与其他

1.初等数论

- 【8】 原根和指数
- 【8】 大步小步 (Baby Step Giant Step, BSGS)
算法
- 【9】 狄利克雷 (Dirichlet) 卷积
- 【10】 二次剩余
- 【10】 二次同余式

2.离散与组合数学

- 【9】 群及其基本性质
- 【9】 置换群与循环群
- 【9】 母函数
- 【9】 莫比乌斯反演
- 【9】 Burnside引理与Pólya定理
- 【9】 斯特林 (Stirling) 数
- 【9】 无根树的Prüfer 序列

3.线性代数

- 【9】 逆矩阵
- 【9】 行列式
- 【9】 向量空间与线性相关
- 【9】 基与线性基

4.高等数学

- 【8】 多项式函数的微分
- 【8】 多项式函数的积分
- 【10】 泰勒 (Taylor) 级数
- 【10】 快速傅里叶变换

5.概率论

- 【8】 概率的基本概念
- 【9】 离散随机变量的期望与方差
- 【9】 条件概率
- 【9】 贝叶斯公式

6. 博弈论

- 【9】 尼姆 (Nim) 博弈
- 【9】 SG函数

7. 最优化

- 【10】 单纯形法

8. 计算几何

- 【8】 点、线、面之间位置关系的判定
- 【8】 一般图形面积的计算
- 【8】 二维凸包
- 【9】 半平面交

9. 信息论

- 【10】 熵、互信息、条件熵、相对熵

10. 其他

- 【10】 信息复杂度的概念
- 【10】 描述复杂度的概念
- 【10】 通讯复杂度的概念

