



# *National Olympiad in Informatics*

全国青少年信息学奥林匹克系列

## **竞赛大纲** (2023 年修订版)

CCF NOI 科学委员会全体审定



中国计算机学会  
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** (2023 年修订版)

CCF NOI 科学委员会全体审定



中国计算机学会  
China Computer Federation

## >>>> 序 <<<<

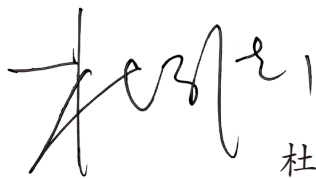
全国青少年信息学奥林匹克竞赛（NOI）于 1984 年创办，至今已经走过了近四十个年头。NOI 从无到有，通过持续发展革新，已经从单一的小规模赛事发展为形式丰富多样、参加人数众多的系列性活动，每年参加的学生达十余万，选拔出的选手在国际赛场上屡创佳绩。从 NOI 活动走出的大批选手，在名校接受了很好的学术训练，毕业后在计算机专业领域大显身手，很多已经成长为技术领军人才，为社会做出了突出的贡献。

然而，随着计算机技术的发展，特别是人工智能技术即将全面深入到我们的日常工作和生活，我们还需要更多的计算机专业优秀人才，这就给 NOI 活动提出了新的挑战和要求。我们清醒地认识到，NOI 活动的普及面还不够广，特别是合格师资的缺口还很大，不少学校无法配备专业的指导教师，这就为活动发展带来了障碍。此外，尽管 NOI 活动已开展多年，但在知识考察的范围方面还缺乏一个明确的规范，其知识边界主要靠组织者和指导教师的既往经验来把握，这就难免会让活动的开展和选手的成长走一些弯路。这当然是我们不愿意看到的。

为确保 NOI 活动的发展和选手的学习有章可循，并尽快培养出更多的胜任师资，CCF NOI 科学委员会与全国 NOI 界数十位指导教师共同努力，于 2021 年制定完成并首次发布了 NOI 大纲。大纲发布后，对竞赛组织、教师教学和选手学习发挥了很好的引导作用，但是也发现了一些不足，需要尽快完善。大纲工作组经过近一年的努力，在广泛吸取各方意见的基础上，完成了本次修订。

我相信这部大纲对 NOI 以及计算机科学普及等活动将会持续发挥积极的引导作用。同时，NOI 科学委员会也秉承 CCF 一向坚持的开放思路，诚挚希望广大指导教师和选手能够在学习和竞赛的过程中，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。

我非常感谢 NOI 科学委员会和相关指导教师对大纲的完善所付出的努力，特别感谢大纲工作组组长赵启阳博士从大纲的起草到修订所做出的重要贡献。



杜子德

全国青少年信息学奥林匹克竞赛（NOI）主席

2023 年 3 月

## 全国青少年信息学奥林匹克系列竞赛大纲

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、彭泱  
（特邀）、钟诚（特邀）

日期：2023-3-9

# >>>> 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、大 纲 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 系列活动开展的现状，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 以及 CCF 主办的其他有关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, 简称 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, 简称 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 (CTS)、NOI 冬令营、国家集训队集训等。

除上述等级以外，还对所有知识点标定了学习难度系数（范围为 1~10）。考虑到相邻级别中知识点的难度系数范围可能互有交叉，入门级知识点难度系数范围取 1~5，（除入门级知识点外的）提高级知识点难度系

数范围取 5~8，（除入门级、提高级知识点外的）NOI 级知识点难度系数范围取 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等级知识点的范围，以尽可能清晰地划定相应等级的知识范围，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等级知识点的范围。为此，大纲在制定中将采取“上粗下细”的指导思想：知识等级越低，其内容规定得越细；知识等级越高，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，高等级比赛的知识范围将自动地包含低等级比赛的所有知识点。同时，对每个等级按照竞赛环境 (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；



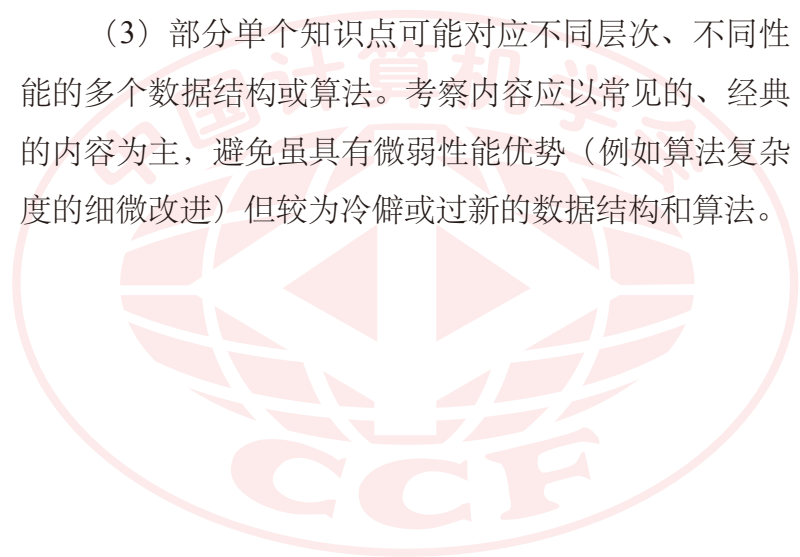
某些知识点可能与多个类别均有紧密或松散联系，本大纲均按其主要属性划定其类别，以避免同一知识点在多个类别中的重复出现。



### 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) 本轮大纲工作组成员为: 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。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, 左骏驰、杨博洋、刘承奥以及多位未具名的师生均提出了各类宝贵的修订意见, 在此表示感谢。

## >>>> 二、大纲 <<<<

### 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】 g++、gcc 等常见编译器的基本使用

### 2.1.2 C++ 程序设计<sup>1</sup>

#### 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

<sup>1</sup> 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. 指针类型

- 【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

## 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】 算法模板库中的函数: 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. 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

## 5. 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

## 6. 搜索算法

- 【5】深度优先搜索
- 【5】广度优先搜索

## 7. 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)

## 8. 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】模运算与同余
- 【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】 格雷码

## 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】 g++、gcc 等编译器与相关编译选项
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<sup>2</sup>

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)

<sup>2</sup> C++ 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】 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 树)
- 【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

## 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. 字符串相关算法

- 【5】字符串匹配：KMP 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

- 【7】次小生成树

- 【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】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<sup>3</sup>

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. 序列

- 【9】 跳跃表

#### 3. 复杂树

- 【8】 树链剖分
- 【10】 动态树: LCT
- 【8】 二维线段树
- 【9】 树套树
- 【9】 k-d 树
- 【10】 虚树

#### 4. 可合并堆

- 【8】 左偏树
- 【10】 二项堆

#### 5. 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】 Manacher 算法
- 【9】 扩展 KMP 算法
- 【8】 有穷自动机
- 【8】 AC 自动机

<sup>3</sup> C++ 以外的其他高级程序设计语言可参照本部分内容。

- 【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
- 【9】 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

### 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
- 【10】 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



noi



中国计算机学会  
China Computer Federation

全国青少年信息学奥林匹克竞赛 (CCF NOI)

Email: [noi@ccf.org.cn](mailto:noi@ccf.org.cn)

网址: <http://www.noi.cn>