



# *National Olympiad in Informatics*

## 全国青少年信息学奥林匹克系列 竞赛大纲

CCF NOI 科学委员会全体审定



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>



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

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）自 1984 年创办至今，已经走过了三十七个年头。NOI 活动从无到有，通过持续发展革新，目前活动形式丰富多样，每年直接参加活动的学生即达十余万，选拔出的选手在国际赛场上屡屡夺金。多年以来，NOI 活动培养的选手已经在计算机专业领域大显身手，成为各单位争相聘用优秀人才。

然而，在辉煌的历史成绩面前，我们决不能故步自封，而是要清醒地认识到，NOI 活动的普及面还不够广，合格师资的缺口还很大，因此需要进一步发展和提高。为使参加 NOI 活动的老师和学生有章可循，CCF NOI 科学委员会推出了这份竞赛大纲，也是想在这方面做出一些促进的工作。

另一方面，尽管 NOI 开展多年，但系列比赛所涉及的知识体系及其边界主要靠教师和学生根据过去的经验来把握，其间要付出更多的辛苦，而多走弯路也是常遇到的事。这在 NOI 活动开展较为滞后的地区尤



甚。如果有一个大纲能够比较明确地给出竞赛的知识范围，那么选手的学习和教师的教学都将更有针对性，效率也会更高。

在 NOI 科学委员会各位委员和全国几十位工作在第一线的 NOI 指导教师的共同努力下，这份大纲终于成稿并可对外发布了。我代表中国计算机学会，对参与大纲编制工作的老师们表示衷心的感谢，特别要感谢主要组织和起草人赵启阳博士，他为这个大纲的成型付出了极大的努力和智慧；感谢科学委员会主席王宏博士及顾问尹宝林教授，他们在许多关键的问题上都给出了真知灼见。

我非常有信心，这份大纲将会对计算机科学普及和 NOI 竞赛发挥积极的引导作用。同时，也请使用这个大纲的老师和同学们多提宝贵的修改意见或建议。



杜子德

中国计算机学会 秘书长  
全国青少年信息学奥林匹克竞赛 (NOI) 主席  
2021 年 1 月

## 全国青少年信息学奥林匹克系列竞赛大纲

CCF NOI 科学委员会全体审定

起草与修订：朱全民、江 涛、宋新波、汪星明、  
胡伟栋、赵启阳

第一轮审阅：韩文骏、邱桂香、李 曙、叶国平、  
滕 伟、蒋 帅、缪和平、洪嘉毅、  
韩 冰

第二轮审阅：陈合力、徐先友、曹利国、史习艺、  
叶金毅、钟 诚、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

特邀审阅：彭 泱

审定日期：2020-12-29

# >>>> 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

### 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



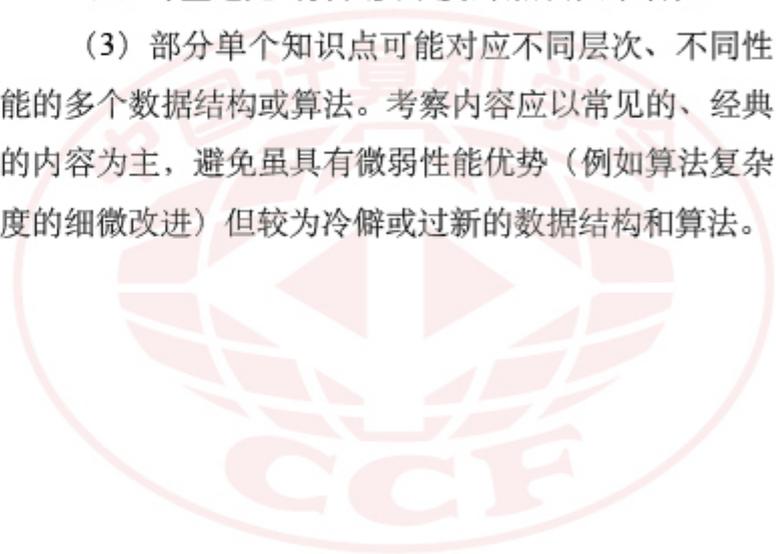
分类。对每个大类又按照知识点的属性继续划分为若干小类；某些知识点可能与多个类别均有紧密或松散联系，本大纲均按其主要属性划定其类别，以避免同一知识点在多个类别中的重复出现。



### 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) 尽量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 的发展而定期进行维护和修订，周期暂定为两年；

(2) 本轮大纲维护小组成员为：朱全民、江涛、宋新波、汪星明、胡伟栋和赵启阳。欢迎将对大纲的修订意见反馈给以上人员。



## >>>> 二、大纲 <<<<<

### 2.1 入门级

#### 2.1.1 计算机基础与编程环境

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】** 计算机的历史及其在现代社会中的常见应用
5. **【1】** NOI 以及相关活动的历史
6. **【1】** 进制的基本概念与进制转换、字节与字
7. **【1】** 程序设计语言以及程序编译和运行的基本概念
8. **【1】** 使用图形界面新建、复制、删除、移动文件或目录
9. **【1】** 使用 Windows 系统下的集成开发环境（例如 Dev C++ 等）
10. **【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)

11. 【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定义与理解
- 【2】 编辑、编译、解释、调试等概念理解

#### 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, 取上整函数, 取下整函数, 常用三角函数, 对数函数, 指数函数, 平方根函数

#### 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】数组定义，数组与数组下标的含义
- 【1】数组的读入与输出
- 【2】纯一维数组的综合运用
- 【3】纯二维数组与多维数组的综合应用

## 8. 字符串的处理

- 【2】字符数组与字符串的关系
- 【2】字符数组的综合应用
- 【2】string 类定义、相关函数引用
- 【3】string 类的综合应用

## 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】递归函数的概念、定义与调用



## 10. 结构体类型

- 【3】结构体的定义及应用

## 11. 指针类型

- 【4】指针的概念及调用
- 【4】指针与数组
- 【4】字符指针与 string 类
- 【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 模板应用

- 【3】<algorithm> 中 sort 函数
- 【4】栈 (stack)、队列 (queue)、链表 (list)、向量 (vector) 等容器

### 2.1.3 数据结构

#### 1. 线性表

- 【3】链表：单链表、双向链表、循环链表



- 【3】 栈
- 【3】 队列
  
- 2. 简单树
  - 【3】 树的定义及其相关概念
  - 【4】 树的父亲表示法
  - 【3】 二叉树的定义及其基本性质
  - 【4】 二叉树的孩子表示法
  - 【4】 二叉树的遍历：前序、中序、后序遍历
  
- 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、构造及其遍历
  - 【4】 二叉排序树的定义、构造及其遍历
  
- 4. 简单图
  - 【3】 图的定义及其相关概念
  - 【4】 图的邻接矩阵存储
  - 【4】 图的邻接表存储



## 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】求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（稳定性等）
- 【3】冒泡排序
- 【3】简单选择排序
- 【3】简单插入排序

## 6. 图论算法

- 【4】图的深度优先遍历算法
- 【4】图的宽度优先遍历算法
- 【5】洪水填充算法（floodfill）

## 7. 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

### 1. 数及其运算

- 【1】数的概念，算术运算（加、减、乘、除、求余）
- 【1】数的进制：二进制、八进制、十六进制和十进制及其转换
- 【2】编码：ASCII 码，哈夫曼编码，格雷码

### 2. 初中数学

- 【1】初中代数
- 【1】初中平面几何

### 3. 初等数论

- 【3】整除、因数、倍数、指数、质数、合数、同余等概念
- 【3】唯一分解定理
- 【3】欧几里德算法（辗转相除法）
- 【4】埃氏筛法和线性筛法求素数

### 4. 组合数学

- 【2】加法原理



- 【2】乘法原理
- 【4】排列及计算公式
- 【4】组合及计算公式
- 【4】杨辉三角公式



## 2.2 提高级

### 2.2.1 计算机基础知识与编程环境

1. 【5】在 Linux 系统终端中使用 `mkdir`、`cp`、`rm`、`mv` 等命令新建、复制、删除、移动文件或目录
2. 【5】在 Linux 系统终端中使用 `cd`、`pwd`、`ls` 等命令更改、显示目录路径和查看目录中的文件
3. 【5】在 Linux 系统下使用 `Gedit`、`Vim` 或 `Emacs` 等文本编辑工具编写代码
4. 【5】熟悉 `g++`、`gcc` 等编译器以及优化、数学库等常见编译选项
5. 【5】在 Linux 系统终端中运行程序，并使用 `time` 命令查看程序用时（区分 `real time`、`sys time` 和 `user time`）
6. 【5】了解调试工具 `gdb` 及其 `break`、`display`、`continue`、`step` 等命令



## 2.2.1 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】集合 (set)
- 【5】列表 (list)，双端队列 (deque)，优先队列 (priority\_queue)
- 【5】多重集合 (multiset)
- 【5】映射 (map)，多重映射 (multimap)
- 【5】对 (pair)，元组 (tuple)

## 2.2.2 数据结构

### 1. 线性结构

- 【5】双端栈
- 【5】双端队列
- 【5】有序队列
- 【6】优先队列
- 【6】倍增表 (ST 表)

<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。



## 2. 集合与森林

- 【6】等价类
- 【6】并查集
- 【6】树与二叉树的转化——孩子兄弟表示法

## 3. 特殊树

- 【6】线段树与树状数组
- 【6】字典树 (trie 树)
- 【7】笛卡尔树
- 【8】二叉平衡树 AVL、treap、splay 等
- 【8】基环树

## 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】重连通图



## 5. 哈希表

- 【5】数值哈希函数构造
- 【6】排列哈希函数构造
- 【6】字符串哈希函数构造
- 【6】哈希函数冲突的常用解决方法

### 2.2.3 算法

#### 1. 复杂度分析

- 【6】空间复杂度分析
- 【6】时间复杂度分析

#### 2. 基础算法

- 【6】分治算法

#### 3. 排序算法

- 【5】归并排序
- 【5】快速排序
- 【6】堆排序
- 【6】树形选择排序（锦标赛排序）
- 【5】桶排序
- 【6】基数排序



## 4. 字符串相关算法

- 【5】字符串匹配算法——KMP

## 5. 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】双向宽度优先搜索
- 【7】迭代加深搜索
- 【8】搜索对象的压缩存储

## 6. 图论算法

- 【6】Prim 和 kruskal 等求最小生成树算法
- 【7】求次小生成树算法
- 【6】Dijkstra、bellman\_ford、SPFA 等求单源最短路算法
- 【7】求单源次短路径算法
- 【6】Floyd-Warshall 算法求任意两点间的最短路和传递闭包
- 【6】有向无环图的拓扑排序算法
- 【6】求欧拉道路和欧拉回路算法



- 【6】二分图的构造及其判定算法
- 【6】最近公共祖先
- 【7】求强联通分量算法
- 【7】强连通分量的缩点算法
- 【7】求割点、割边算法

### 7. 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

## 2.2.4 数学

### 1. 高中数学

- 【5】代数
- 【6】解析几何
- 【6】立体几何

### 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】逆元
- 【7】扩展欧几里得算法
- 【7】孙子定理（即中国剩余定理）

### 3. 组合数学

- 【6】可重集排列
- 【6】可重集组合
- 【6】错排列、圆排列
- 【6】鸽巢原理
- 【6】二项式定理
- 【7】容斥原理
- 【7】卡特兰数

### 4. 线性代数

- 【5】矩阵概念
- 【6】特殊矩阵：稀疏矩阵，三角矩阵，对称矩阵
- 【6】矩阵的初等变换
- 【6】矩阵的加减乘和转置运算
- 【7】线性方程组的高斯消元法



## 2.3 NOI 级

### 2.3.1 C++ 程序设计<sup>3</sup>

1. **【8】** STL 模板: 容器 (containers)、迭代器 (iterators)、空间配置器 (allocators)、配接器 (adapters)、算法 (algorithms)、仿函数 (functors)
2. **【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】** 分块
- **【8】** 块状链表

#### 2. 序列

- **【8】** 后缀数组
- **【9】** 跳跃表
- **【9】** 无根树的 Prüfer 序列

#### 3. 复杂树

- **【8】** 树链剖分

<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】** 主席树
- **【8】** 二维线段树
- **【9】** 后缀树
- **【9】** 树套树
- **【9】** k-d 树
- **【10】** 最小树形图
- **【10】** 动态树 (LCT)

#### 4. 可合并堆

- **【8】** 左偏树
- **【10】** 二项堆

#### 5. **【9】** 可持久化数据结构

### 2.3.3 算法

#### 1. 算法策略

- **【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】** 求最长回文串的 Manacher 算法



- 【8】多模匹配算法——AC 自动机
- 【9】求字符串前缀和后缀算法——扩展 KMP
- 【9】确定性有穷自动机——DFA 算法
- 【10】非确定性有穷自动机——NFA 算法
- 【10】后缀自动机

### 3. 图论算法

- 【8】网络流算法
- 【10】图的支配集、独立集与覆盖集
- 【8】二分图的最大匹配——匈牙利算法
- 【9】二分图的最佳匹配算法——KM 算法
- 【10】一般图的匹配

### 4. 动态规划

- 【9】复杂动态规划模型构建
- 【9】复杂动态规划模型的优化

## 2.3.4 数学

### 1. 信息论基础

- 【10】熵、互信息、条件熵、相对熵的基本概念
- 【10】信息复杂度的基本概念



- 【10】描述复杂度的基本概念
- 【10】通讯复杂度的基本概念

### 2. 初等数论

- 【8】原根和指数
- 【8】大步小步 (Baby Step Giant Step, BSGS) 算法
- 【9】完全数
- 【9】狄利克雷 (Dirichlet) 卷积
- 【10】平方剩余
- 【10】二次同余式
- 【10】二次互反律

### 3. 离散数学

- 【9】代数系统的基本概念
- 【9】群的基本概念
- 【9】置换群与循环群

### 4. 组合数学

- 【9】母函数
- 【9】莫比乌斯变换
- 【9】Burnside 引理与 Pólya 原理



- 【9】斯特林数

## 5. 高等数学

- 【9】多项式函数微分
- 【9】多项式函数积分
- 【10】泰勒级数
- 【10】快速傅里叶变换 (Fast Fourier Transform, FFT)
- 【10】卷积

## 6. 线性代数

- 【9】矩阵的逆运算
- 【9】行列式及其运算
- 【9】线性相关与矩阵的逆

## 7. 概率论

- 【8】概率相关概念
- 【9】求概率的乘法公式、全概率公式、贝叶斯公式

## 8. 博弈论

- 【9】零和博弈问题——Nim 博弈等



- 【9】Sprague-Garundy (SG) 函数概念及应用

## 9. 运筹学

- 【10】线性规划之单纯形法

## 10. 计算几何

- 【7】矢量及其运算
- 【8】点、线、面之间的位置判断
- 【8】常见图形的面积计算
- 【8】二维凸包的求法及其应用
- 【9】半平面交

