

第十五届广东省青少年人工智能与机器人大赛 AIOJ 算法挑战规则（小学低龄组、高龄组、中学组）

1. 大赛简介

(1) 公平、公正、公开

采用 OI/ACM 赛制+滚榜机制，设立实时滚动榜单将比赛成绩排名用大屏幕对外即时公开，任何人都可进行查看。

(2) 统一由 AIOJ 系统评测

无论是图形化还是 C++ 语言，无论是客观题还是编程题全部使用先进的 AIOJ 进行自动评测，无需人工阅卷，赛后快速出成绩。

(3) 科学命题，接轨信奥赛

由命题专家专业统筹命题，聚焦算法思维，场景化设计、项目化设计，难度均衡，考察合理。同时与信奥赛体系考察内容保持高度一致。

(4) 赛制包容，出口众多

客观真实的比赛成绩可作为编程能力认证、信息学大赛等检验学生算法与编程能力的重要参考依据。

2. 大赛组别

序号	赛项	小低组 (1-3年级)	小高组 (4-6年级)	中学组
1	计算思维 (无需编程)	√	√	×
2	图形化语言	√	√	×
3	Python语言	×	√	√
4	C++语言	×	√	√

3. 赛程安排

(1) 比赛内容

赛项类型	选手组别	年龄要求	时长	题型	题量	分值
计算思维 (无需编程)	小低组	小学 1-3年级	60分钟	单选题	10	6
				填空题	5	8

	小高组	小学 4-6年级	60分钟	单选题	10	4
				填空题	5	6
				多选题	5	6
图形化 算法	小低组	小学 1-3年级	60分钟	单选题	15	4
				算法编程题	1	40
	小高组	小学 4-6年级	90分钟	算法分析题	1	30
				算法编程题	2	35
Python 算法	小高组	小学 4-6年级	90分钟	算法分析题	2	30
				算法编程题	1	40
	中学组	7-9年级 学段	90分钟	算法分析题	1	30
				算法编程题	2	35
C++ 算法	小高组	小学 4-6年级	90分钟	算法分析题	2	30
				算法编程题	1	40
	中学组	7-9年级 学段	90分钟	算法分析题	1	30
				算法编程题	2	35

(2) **比赛形式：**在线答题。

比赛由 AIOJ (aioj.net) 系统自动评测，学生凭主办单位要求的账号、密码登录考场电脑的测评系统（赛前统一公布），单人闭卷作答，在网页上提交答案。

(3) **比赛地点：**由主办单位安排组织线上进行。

(4) **参赛语言说明：**所有参赛语言均为通用版本，如【图形化算法】为 Scratch3.0。

4. 报名办法

(1) 报名时以【个人】形式参赛。

(2) 每位参赛学生仅填写 1 位指导教师。

(3) 以参赛对象所在学校（含民办）、科技馆、少年宫、青少年活动中心、培训机构等为单位组织学生进行团体报名（本比赛不接受个人报名）。本赛项官方客服咨询微信：aioj03。

5. 奖励设置

本次比赛将按照如下方式进行评奖并颁发证书：

（1）初赛不设评奖。如果有地市主办单位，将由其对所在地市参赛学生单独进行评奖。

（2）决赛获奖比例：一等奖约占比 15%，二等奖约占比 35%，三等奖约占比 50%。

6. 联系方式

本赛项联系人：章老师 咨询电话：18922852573

7. 其他说明

（1）本规则由广东实践创新教育研究院负责制定解释。规则如有调整，将在省赛前公布定稿规则。

（2）本规则是实施裁判工作的依据，裁判长对规则中未说明事项以及有争议事项，均拥有最后解释权和最终裁定权。选手必须在赛后（未离开竞赛区前）提出异议，否则视为无异议。

（3）本规则坚持青少年科技教育公益性和资源共建共享的原则，公开免费下载使用，不作商业用途。使用该规则时，须注明规则来源，亦不得损害规则制定方的有关权益。

附件

「AIOJ」青少年算法能力挑战赛考察范围

一、【计算思维】考察范围

1. 整体要求

本次考察范围将以《义务教育信息科技课程标准（2022 年版）》为基础，主要考查如下几方面学生对于计算思维和算法思维的理解应用能力：

问题分解：将复杂的现实问题或系统拆解为规模更小、功能更单一、相互独立且可协作解决的子问题（或模块）的思维过程。。

抽象建模：从具体问题情境中提取关键属性与关系，过滤无关细节，构建形式化或半形式化的概念模型，以表达问题本质结构的思维过程。**算法设计：**针对已分解和建模后的问题，设计明确的、有穷的、机械化的操作步骤序列，以在有限资源约束下完成特定任务或达成目标状态的过程。

评估迁移：对设计的解决方案进行有效性、效率性、鲁棒性（容错性）的系统性检验与反思，并将经过验证的思维模式、方法策略或抽象模型应用于新情境或更广泛问题域的能力。

2. 具体到各组别的考核侧重点如下：

组别	核心目标	核心考察范围
小低组 (1-3年 级)	考察学生对简单任务的顺序感知、基础逻辑判断和逻辑推理能力，能识别简单流程、用直观方式表达步骤，初步建立“按步骤解决问题”的思维意识。	1. 任务顺序与逻辑感知：能准确判断生活中简单任务的先后逻辑顺序，清晰区分任务执行的先后层级，能排除不合理的步骤排序；同时能根据简单条件做出“是/否”“能/不能”的基础逻辑判断，初步感知“条件匹配”这一核心计算思维基础思想。 2. 图符与流程认知：能识别简单流程符号（如箭头、开始/结束标记），看懂用图符绘制的生活化简单流程图，并能根据流程图完整、准确说出任务执行顺序，培养流程化思维意识。 3. 分类整理与场景应用：能根据给定标准（如颜色、形状、用途）对物品进行规范分类，清晰阐述分类依据，感知“分类”这一基础计算思维方法；结合校园、家庭等真实生活场景（如整理玩具、安排课间活动），能梳理并说出解决问题的2-3个关键步骤，初步建立“分解任务”的思维意识，体现“做中学、用中学”的

		<p>课标理念。</p> <p>4. 能简单运用排除法、倒推法、假设法进行逻辑推理。</p>
小高组 (4-6年 级)	<p>考察学生的问题分解能力、简单算法设计能力，能运用自然语言或流程图描述算法，理解“顺序、分支”等基础算法思想，能解决简单的生活和数学中的思维问题。</p> <p>考察学生的抽象建模、算法设计与优化能力，能运用计算思维分解复杂问题、设计完整算法，理解“循环”等进阶算法思想，能解决稍复杂的跨场景问题，具备初步的思维迁移和优化意识。</p>	<p>1. 问题分解：能将稍复杂任务分解为3-4个可执行子任务，明确每个子任务的核心目标，不遗漏关键步骤，培养“化繁为简”的思维能力。</p> <p>2. 基础算法思想（顺序、分支）：落实课标“用自然语言或流程图描述简单算法”的要求，具体包括：顺序思想：能设计“先做→再做→最后做”的连贯步骤，解决煮面条、制作手工等具体生活问题，并能清晰、有条理地表达步骤逻辑；分支思想：能根据不同条件选择差异化解决步骤，能准确识别并补充分支流程图的缺失部分，建立“条件对应结果”的逻辑思维。</p> <p>3. 流程图应用：能看懂含顺序、分支符号的完整简单流程图，能根据题目要求，运用开始/结束、步骤、箭头、判断等简单流程图符号，绘制对应算法的流程图。</p> <p>4. 模式识别与简单推理：能快速发现数字、图形的简单规律（如等差数列、图形轮换），并根据规律推理出后续结果；能运用排除法等简单推理方法，解决找物品、简单数独入门等问题，培养规律感知与逻辑推理能力。</p> <p>5. 数据整理与简单分析：能对班级同学兴趣爱好、日常开销等少量生活化数据进行分类、计数，能根据整理结果提炼简单结论（如“喜欢跳绳的同学最多”），落实课标“数据处理”的基础要求。</p> <p>6. 场景化思维应用：结合校园、家庭等真实生活场景，能设计简单解决方案，并能用自然语言或流程图清晰描述解决步骤。</p> <p>7. 能简单运用排除法、倒推法、假设法、数理思维进行逻辑推理。</p> <p>8. 抽象与建模：能忽略问题无关细节、抓住核心特征，运用简单符号、图表或文字建立问题模型，培养抽象思维能力。</p> <p>9. 进阶算法思想（循环、优化）：落实课标“设计完整算法、优化解决方案”的要求，具体包括：循环思想：能准确理解“重复执行某一步骤直到满足条件”的核心逻辑，能识别、补充循环流程图，能独立设计含循环步骤的算法，掌握进阶算法思维基础；优化思想：能对比不同解决方案的优劣，筛选出更简洁、高效的</p>

		<p>解决方法（如“最省时间的做事顺序”），并能清晰阐述优化理由。</p> <p>10. 算法设计与表达：能针对校园图书角管理、运动会赛程安排等稍复杂问题，设计完整算法流程，能用规范流程图或自然语言清晰、有条理地描述算法步骤，确保逻辑无漏洞，落实课标“算法设计与表达”的核心要求。</p> <p>11. 逻辑推理与复杂规律：能完成复杂数独、推理谜题等难度较高的逻辑推理题目，能发现数字、图形的复杂规律并进行多步推理；能理解简单“因果逻辑”，分析问题产生的根源，强化逻辑思维深度。</p> <p>12. 数据处理与分析：能对较多生活化数据进行分类、排序、统计，能运用计算平均数、对比百分比等简单方法分析数据，能根据分析结果提出合理建议。</p> <p>13. 跨场景思维迁移：能将解决某一问题的计算思维方法迁移到类似新场景中（如将“整理图书的方法”迁移到“整理错题本”）；能结合数学等其他学科知识，解决购物省钱方案等跨学科简单问题。</p> <p>14. 简单编程逻辑感知（非代码编写）：能理解积木式编程的基本逻辑（如角色移动、条件判断、循环执行），能根据编程逻辑预测执行结果，或根据预期结果补充编程逻辑步骤，不考察具体代码编写。</p>
--	--	--

二、【图形化算法】考察范围

核心能力要求	核心考察范围	小学低年级组	小学高年级组
1. 具备图形化编程工具的应用能力，掌握图形化编程基础，掌握三大基本结构，能够独立完成基本作品。	<p>计算机基础知识，计算机历史，图形化编程平台基本操作；</p> <p>背景/角色的添加和切换，角色的属性及控制，声音模块；</p> <p>触发事件，侦测条件，多角色设置；</p> <p>三大基本结构。</p>	√	√

2. 理解编程相关抽象概念，具备一定的逻辑推理能力及基础设计能力，能够独立完成包含分支语句、循环语句等比较综合的案例。	计算机网络和Internet基本概念，流程图的概念与描述，克隆，画笔拓展模块；算术运算，广播；变量，逻辑推理和编程数学(逻辑运算)。	√	√
3. 熟悉图形化编程高级工具使用并能进行综合运用，能够实现知识迁移，通过编程解决生活中的问题。	程序输入与输出，字符串处理。 克隆的综合应用，复杂的嵌套结构； 复杂的逻辑判断，算法概念与描述。 算法的基础运用：简单枚举法；简单模拟法；	√	√
4. 具备计算思维，能够通过分层、抽象、模式识别、设计流程解决实际问题。	函数，列表的增删改查； 算法的复杂运用； 排序概念和稳定性；经典排序(冒泡排序、插入排序、选择排序等)综合应用。	X	√

三、【Python 算法 和 C++ 算法】考察范围相同

核心能力要求	核心考察范围	小学高年级组	中学组
1. 能熟练操作集成开发环境（创建 / 编辑 / 编译 / 调试文件），正确使用头文件与标准命名空间； 2. 能规范定义使用常量变量，完成基础算术 / 逻辑 / 关系 / 复合运算； 3. 能独立编写含顺序、分支、循环结构的简单程序，会使用循环终止 / 继续语句。	计算机基础与编程环境，计算机历史； 变量的定义与使用；基本数据类型（整型、浮点型、字符型、布尔型）控制语句结构（顺序、循环、选择）； 基本运算（算术运算、关系运算、逻辑运算），输入输出语句。	√	√
1. 能绘制流程图并描述程序三大基本结构，实现 ASCII 码与字符的相互转换；	计算机的存储与网络程序设计语言的特点流程图的概念与描述ASCII 编码； 数据类型的转换；多层分支/循环结构	√	√

<p>2. 能灵活进行强制 / 隐式数据类型转换，独立编写含多层分支 / 循环嵌套的程序；</p> <p>3. 能熟练使用基础数学函数解决简单计算问题。</p>	<p>；</p> <p>常用数学函数（绝对值函数、平方根函数、max 函数、min 函数）。</p>		
<p>1. 能完成不同进制相互转换，实现基本位运算；</p> <p>2. 能用自然语言 / 流程图 / 伪代码描述算法；</p> <p>3. 能熟练处理一维结构化数据，完成字符串的各类通用操作；</p> <p>4. 能独立使用枚举法、模拟法构造算法、描述算法并实现调试，解决实际算法问题。</p>	<p>数据编码（原码、反码、补码）；</p> <p>进制转换（二进制、八进制、十进制、十六进制）；</p> <p>位运算（与（&）、或（ ）、非（~）、异或（^）、左移（<<）、右移（>>））；</p> <p>算法的概念与描述（自然语言描述、流程图描述、伪代码描述）；</p> <p>一维数组基本应用，字符串及其函数；</p> <p>字符串处理：string类核心常用函数；</p> <p>算法： 枚举法，模拟法；</p> <p>排序算法：计数排序的原理与代码实现；</p> <p>STL 基础：向量容器vector的基础使用（定义、增删改查、遍历）；</p> <p>算法策略：一维前缀和、一维差分的概念与基础实现。</p>	√	√
<p>1. 能推导递推关系式，使用递推算法实际问题；</p> <p>2. 能实现冒泡 / 插入 / 选择 / 桶排序，理解排序算法的稳定性、复杂度及使用场景；</p> <p>3. 能完成文件重定向与读写操作，处理程序基础异常；</p> <p>4. 能估算多项式 / 指数级别的算法复杂度，根据场景选择最优算法。</p>	<p>函数的定义与调用、形参与实参、作用域；函数参数传递的概念；</p> <p>二维数组与多维数组基本应用；</p> <p>算法：递推；算法：排序概念和稳定性；</p> <p>算法：排序算法（冒泡排序、插入排序、选择排序）简单算法复杂度的估算（含多项式、指数复杂度）；</p> <p>STL 基础：栈stack、队列queue的基础使用；</p> <p>二维数组应用：关系矩阵（0-1 关联矩</p>	√	√

	<p>阵的存储与简单统计，无图论语义）；</p> <p>文件重定向与文件读写操作异常处理；</p> <p>文件操作：二进制文件的基础概念与简单读写操作；</p> <p>算法策略：二维前缀和、二维差分的基础实现。</p>		
<p>1. 能运用初等数论知识解决数论相关问题；</p> <p>2. 能使用二分查找 / 二分答案算法解决实际问题，设计实现递归程序并分析其复杂度；</p> <p>3. 能运用归并 / 快速排序完成数据排序，使用贪心算法解决符合最优子结构的问题；</p> <p>4. 能估算多项式 / 指数 / 对数级别的算法复杂度，了解递归优化策略；</p> <p>5. 能根据实际情况选择合适算法，完成指定功能的实践。</p>	<p>初等数论；</p> <p>数组模拟高精度加法、减法、乘法、除法单链表、双链表、循环链表；</p> <p>辗转相除法（也称欧几里得算法）；</p> <p>素数表的埃氏筛法和线性筛法；唯一分解定理；</p> <p>二分查找/二分答案（也称二分枚举法）；</p> <p>贪心算法；分治算法（归并排序和快速排序）；</p> <p>递归算法复杂度的估算（含多项式、指数、对数复杂度）；</p> <p>STL 进阶：链表容器list、优先队列priority_queue的基础使用；</p> <p>排序算法：堆排序、桶排序、基数排序的原理与实现；</p> <p>数据结构：图的邻接表存储方式（基于链表，仅做数据结构实现，不引入图论语义）；</p> <p>算法策略：离散化算法的基础概念与实现；</p> <p>高精度运算：高精度整数除以单精度整数的商和余数求解。</p>	X	√