

CSP-J 备考：按大纲查缺补漏合集

第一部分 基础知识

<https://mp.weixin.qq.com/s/jEyRWnvCcleIV6N1iTscwA>

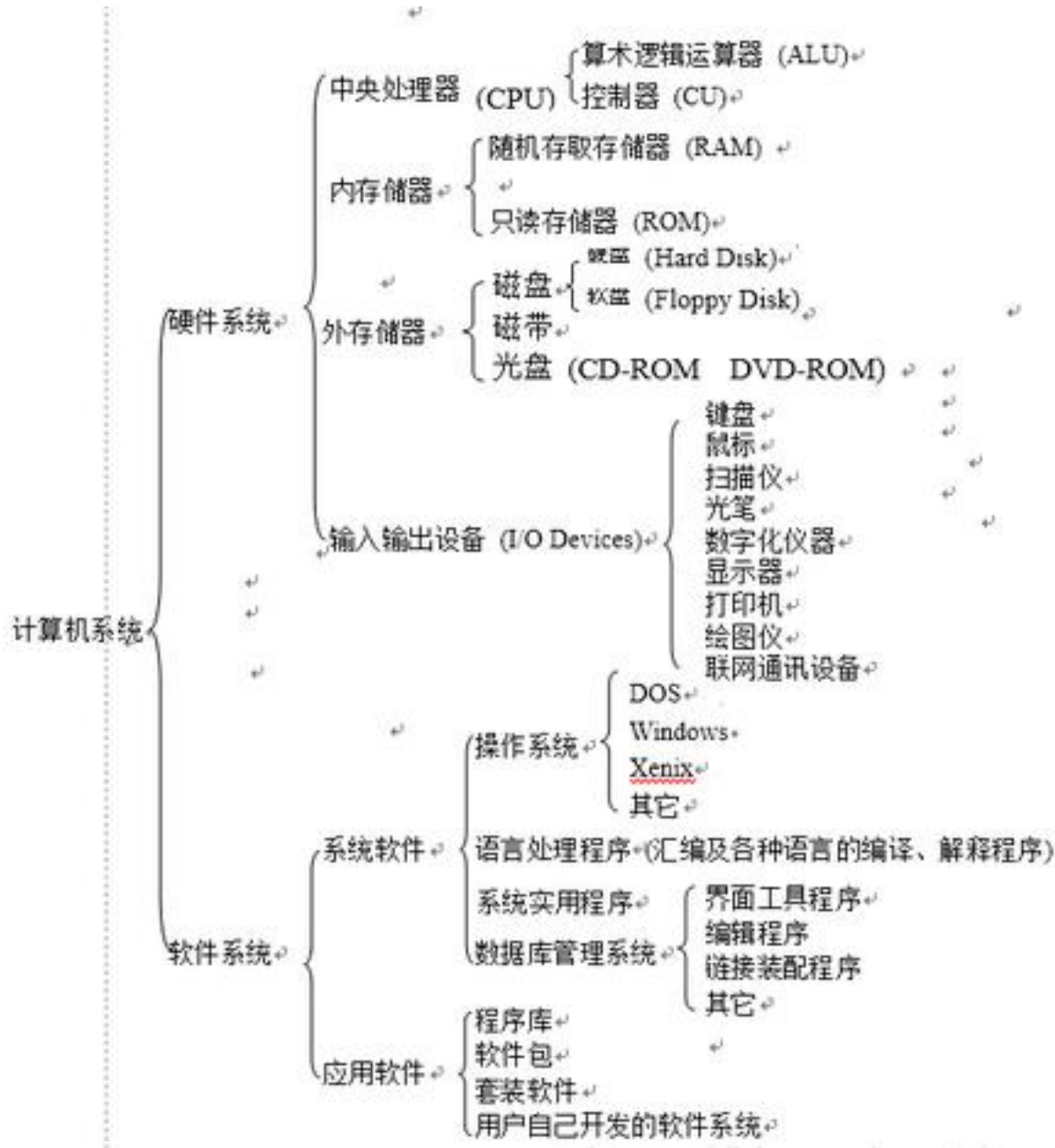
大纲要求

一、基础知识和编程环境

- 1.[1]计算机的基本构成(CPU、内存、IO 设备)
- 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(例如 Code::Blocks 等)
- 12.[1] g++、gcc 等常见编译器的基本使用

简要讲解

1. 计算机的基本构成：计算机系统由硬件和软件两部分组成。计算机硬件部分由中央处理器（CPU）、内存和输入输出（I/O）设备组成。CPU 是计算机的核心部件，负责执行指令和处理数据。内存用于存储数据和程序。I/O 设备包括键盘、鼠标、显示器、打印机等，用于与计算机进行交互。



2. **操作系统**：操作系统是计算机系统的核心软件，负责管理计算机的资源和提供用户与计算机的接口。Windows 和 Linux 是两种常见的操作系统。操作系统提供了文件管理、进程管理、设备管理、用户界面等功能，用户可以通过操作系统进行文件操作、程序运行和系统配置等操作。

3. 计算机网络和 Internet: 计算机网络是指多台计算机通过通信设备相互连接, 实现信息交换和资源共享的系统。Internet 是全球最大的计算机网络, 连接了世界各地的计算机和网络设备。计算机网络和 Internet 提供了数据传输、远程访问、电子邮件、网页浏览等功能。
4. 计算机的历史和常见用途: 计算机起源于 20 世纪 40 年代, 经历了多个发展阶段。计算机的常见用途包括科学计算、数据处理、信息管理、娱乐等。计算机在各个领域都有广泛应用, 如教育、医疗、交通、通信等。
5. NOI 以及相关活动的历史: NOI 是全国青少年信息学奥林匹克竞赛的简称, 是中国的一项信息学竞赛活动。它的历史可以追溯到 1984 年, 是为了培养和选拔青少年信息学人才而举办的。
6. NOI 以及相关活动的规则: NOI 竞赛有一套具体的规则和流程, 包括报名、初赛、复赛和决赛等环节。参赛选手需要通过编程和算法设计的考试来展示自己的能力, 最终获得奖项和荣誉。
7. 位、字节与字: 计算机中最小的存储单位是位 (bit), 它代表一个二进制的 0 或 1。字节 (byte) 是计算机中常用的存储单位, 由 8 个位组成。字是更大的存储单位, 可以由多个字节组成, 字的大小可以根据计算机的架构而变化。
8. 程序设计语言以及程序编译和运行的基本概念: 程序设计语言是一种用于编写计算机程序的语言。编程语言分低级语言 (汇编语言) 和高级语言。常见的高级语言有 C、C++、Java、Python 等。编译是将程序源代码转换为可执行代码的过程, 而运行是指计算机执行程序并产生相应的结果。高级语言比低级语言更容易移植。
9. 使用图形界面新建、复制、删除、移动文件或目录: 在图形界面操作系统中, 用户可以通过鼠标和图形界面进行文件和目录的操作。新建文件或目录、复制文件或目录、删除文件或目录、移动文件或目录是常见的文件操作。
10. 使用 Windows 系统下的集成开发环境: Dev C++ 是一种集成开发环境 (IDE), 用于开发 C 和 C++ 程序。它提供了代码编辑、编译、调试等功能, 方便程序员进行开发。
11. 使用 Linux 系统下的集成开发环境: Code::Blocks 是一种在 Linux 系统下常用的集成开发环境。它支持多种编程语言, 如 C、C++ 等, 提供了代码编辑、编译、调试等功能。
12. g++、gcc 等常见编译器的基本使用: g++ 和 gcc 是常见的 C++ 和 C 编译器。编译器将源代码转换为可执行代码。通过命令行或集成开发环境, 可以使用编译器来编译和运行程序。

补充知识

1、冯·诺依曼（美籍匈牙利数学家）结构是指一种计算机体系结构，也被称为**存储程序计算机**。它是由冯·诺依曼于 1945 年提出的，被广泛应用于现代计算机系统中。冯·诺依曼结构的核心思想是将数据和指令存储在同一个存储器中，以及使用存储程序的概念。这种结构包括中央处理器（CPU）、存储器、输入/输出设备 and 控制单元等组件。

冯·诺依曼理论是指冯·诺依曼对计算机科学的一系列理论贡献。他提出了存储程序计算机的概念，推动了计算机科学的发展。冯·诺依曼还提出了计算机的基本原理，如指令的顺序执行、程序的存储和控制等，为计算机科学奠定了基础。

2、图灵是指英国数学家和计算机科学家阿兰·图灵（Alan Turing）。他被认为是现代计算机科学的奠基人之一。图灵在 20 世纪 30 年代和 40 年代做出了许多重要的贡献，尤其是他的图灵机理论，为计算机科学的发展奠定了基础。

图灵奖是由国际计算机协会（ACM）设立的最高荣誉奖项，以纪念图灵的贡献。该奖项每年颁发给在计算机科学领域做出杰出贡献的个人或团队。获得图灵奖被视为计算机科学界的最高荣誉之一。

3、信息论之父是指美国数学家克劳德·香农（Claude Shannon）。他在 20 世纪 40 年代提出了信息论，被誉为信息论之父。信息论是研究信息传输、存储和处理的数学理论，对于通信和数据压缩等领域具有重要意义。

克劳德·香农的信息论研究奠定了通信系统的基础，提出了信息熵、信道容量等重要概念，为信息传输和存储的优化提供了理论依据。他的贡献对于现代通信技术和数据处理具有深远的影响。

第二部分 C++

<https://mp.weixin.qq.com/s/8xVVoxBxd6oxpwByazjWw>

第二部分的必知必会只需要记住粗体字的内容，其它内容通过写代码自然而然就记住了。

大纲要求

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

(11 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

[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(I)、非(~)异或 (^) 、左移 (<<) 、右 (>>)

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

简要讲解

1. 程序基本概念:

- 标识符: 在程序中用来表示变量、函数、类等名称。
- 关键字: 在编程语言中具有特殊含义的单词, **不能作为标识符使用**。

2. 基本数据类型:

- 整数型: 用来表示整数, 如 int 和 long long。**整数型注意数据范围, 记住两个数字 $10^9, 10^{18}$** 。竞赛中不确定的情况下尽量用 long long。
- 实数型: 用来表示实数(包括小数和科学计数法表示的数), 如 float 和 double。
- 字符型: 用来表示单个字符, 如 char。
- 布尔型: bool, 用来表示真或假, 只有两个取值: true 和 false。

3. 程序基本语句:

- 输入输出语句: 用来读取用户输入和输出结果, 如 cin、cout、scanf 和 printf。

- 赋值语句：用来给变量赋值。
- 控制流语句：用来控制程序的执行流程，如 if 语句、switch 语句、for 循环、while 循环和 do while 循环。
- 复合语句：由多条语句组成的语句块。

4. 基本运算：

- 算术运算：加、减、乘、除、整除和求余。
- 关系运算：大于、大于等于、小于、小于等于、等于和不等于。
- 逻辑运算：与（&&）、或（||）、非（!）。逻辑运算还要记数理逻辑运算对应的符号。
- 变量自增与自减运算：对变量进行加 1 或减 1 操作。特别注意前运算和后运算的区别！
- 三目运算：根据条件的真假选择不同的值。
- 位运算：与（&）、或（|）、非（~）、异或（^）、左移（<<）和右移（>>）。
- 赋值运算：+=, -=,...，注意优先级最低，单目运算优先级最高。

记住优先级顺序从高到低：单目-成员>算术>移位，逻辑对比>按位与>异或>或，逻辑与>或>三目>赋值和复合赋值>逗号运算。

5. 数学库常用函数：

- 绝对值函数：abs, 返回一个数的绝对值。
- 四舍五入函数：round, 将一个数四舍五入到最接近的整数。
- 下取整函数：floor, 返回小于或等于一个数的最大整数。
- 上取整函数：ceil, 返回大于或等于一个数的最小整数。
- 平方根函数：sqrt, 返回一个数的平方根。
- 常用三角函数：包括正弦、余弦和正切等函数。
- 对数函数：log, 返回一个数的对数。
- 指数函数：pow, 返回一个数的指数。

6. 结构化程序设计：

- 顺序结构：按照代码的顺序执行。
- 分支结构：根据条件选择不同的执行路径，如 if 语句和 switch 语句。
- 循环结构：根据条件重复执行一段代码，如 for 循环、while 循环和 do while 循环。
- 自顶向下、逐步求精的模块化程序设计：将程序分解为多个模块，每个模块负责完成特定的功能。
- 流程图：用图形化的方式表示程序的执行流程。

7. 数组：

- 数组与数组下标：一组相同类型的数据元素组成的集合，通过下标访问数组中的元素。
- 数组的读入与输出：将数据读入数组或将数组输出。用 for 循环。
- 二维数组与多维数组：由多个一维数组组成的数组。

8. 字符串的处理：

- 字符数组与相关函数：用字符数组表示字符串，并使用相关函数进行字符串的处理。
- string 类与相关函数：C++中提供的字符串类型，可以方便地进行字符串的操作。

9. 函数与递归：

- 函数定义与调用、形参与实参：定义函数并通过传递参数调用函数。
- 传值参数与传引用参数：传递参数时是通过值传递还是通过引用传递。
- 常量与变量的作用范围：变量在程序中的可见范围。
- 递归函数：函数调用自身的函数。

10. 结构体与联合体：

- 结构体：用来存储不同类型的数据成员的自定义数据类型。
- 联合体：一种特殊的数据类型，可以在同一内存空间中存储不同类型的数据。定义方法和用法类型于结构体，但使用时可能会引起数据不正确。

11. 指针类型：

- 指针：存储变量的内存地址。可用取址运算获得指针。从指针获取数值用取值运算。
- 基于指针的数组访问：通过指针访问数组中的元素。
- 字符指针：指向字符的指针。
- 指向结构体的指针：指向结构体类型的指针。注意指针成员->运算符的使用。

12. 文件基本读写：

- 文件的基本概念：计算机中存储数据的一种方式。
- 文本文件的基本操作：读取和写入文本文件的内容。
- 文本文件类型与二进制文件类型：文件的不同存储格式。
- 文件重定向、文件读写等操作：改变标准输入输出流的方向，进行文件的读写操作。

13. STL 模板：

- 算法模板库中的函数：包括最小值、最大值、交换和排序等函数。
- 容器：包括栈、队列、链表和向量等数据结构。

第三部分 数据结构

<https://mp.weixin.qq.com/s/yRq69SwDVCWs27a70OZ-yA>

本次的必知必会内容也不必背记，哈夫曼编码要时不时地练习一下推导和编程。

大纲要求

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

简要讲解

数据结构是计算机科学中非常重要的一个领域，它涉及到如何组织和存储数据以便有效地使用和操作。信息学奥赛大纲中对数据结构进行了详细的划分，包括线性结构、简单树、特殊树和简单图等内容。

1. 线性结构

线性结构是指数据元素之间存在一对一的关系，也就是除了第一个和最后一个数据元素之外，其它数据元素都是首尾相连的。信息学奥赛大纲中要求掌握的线性结构包括链表、栈和队列等。

- 链表

链表是一种常见的数据结构，它由一系列节点组成，每个节点包含数据和指向下一个节点的指针。根据节点之间的连接方式，链表可以分为单链表、双向链表和循环链表。单链表中每个节点只有一个指针指向下一个节点，双向链表中每个节点有两个指针，分别指向前一个节点和后一个节点，而循环链表的最后一个节点指向第一个节点，形成一个循环。

- 栈

栈是一种后进先出（LIFO）的数据结构，只允许在栈顶进行插入和删除操作。插入操作称为入栈，删除操作称为出栈。栈通常用于实现函数调用、表达式求值等场景。

- 队列

队列是一种先进先出（FIFO）的数据结构，允许在队列的一端进行插入操作，另一端进行删除操作。插入操作称为入队，删除操作称为出队。队列常用于实现广度优先搜索、任务调度等场景。

2. 简单树

树是一种非线性数据结构，由节点和边组成，每个节点最多有一个父节点和多个子节点。

- 树的定义与相关概念

树是由节点和边组成的数据结构，其中有且只有一个节点没有父节点，这个节点称为根节点。除了根节点外，每个节点都有且仅有一个父节点。

- 树的表示与存储

树可以通过数组、链表等方式进行表示与存储，常见的方式包括双亲表示法、孩子表示法、孩子兄弟表示法等。

- 二叉树的定义与基本性质

二叉树是每个节点最多有两个子节点的树结构，它具有左子树和右子树之分。

- 二叉树的表示与存储

二叉树可以通过顺序存储结构（数组）或链式存储结构（指针）进行表示与存储。

- 二叉树的遍历

二叉树的遍历包括前序遍历、中序遍历和后序遍历，分别是指先访问根节点、中间节点和后访问根节点。

3. 特殊树

特殊树是指在树的基础上具有特殊性质的树结构，信息学奥赛大纲中要求掌握完全二叉树、哈夫曼树和二叉搜索树。

- 完全二叉树的定义与基本性质

完全二叉树是一种特殊的二叉树，除了最后一层外，每一层的节点数都达到最大值，最后一层的节点依次从左到右排列。

- 完全二叉树的数组表示法

完全二叉树可以通过数组进行表示，利用数组下标之间的关系来表示完全二叉树中节点之间的父子关系。

- 哈夫曼树的定义和构造、哈夫曼编码

哈夫曼树是一种带权路径长度最短的树，常用于数据压缩领域，构造哈夫曼树的过程称为**哈夫曼编码**。（哈夫曼编码经常考到，需要掌握手工推导和编程的方法。参考我写的文章：[逆向思维理解哈夫曼（Huffman）编码](#)）

- 二叉搜索树的定义和构造

二叉搜索树是一种特殊的二叉树，对于每个节点，其左子树上所有节点的值均小于该节点的值，右子树上所有节点的值均大于该节点的值。

4. 简单图

图是一种非线性的数据结构，由节点和边组成。

- 图的定义与相关概念

图由节点和边组成，节点之间通过边相连，边可以有权重，图可以分为有向图和无向图。

- 图的表示与存储: 邻接矩阵

邻接矩阵是一种表示图的方式，通过二维数组来表示节点之间的连接关系。

- 图的表示与存储: 邻接表

邻接表是另一种表示图的方式，通过链表或数组来表示节点之间的连接关系，对于每个节点，记录其相邻节点的信息。

以上是信息学奥赛大纲中关于数据结构部分的内容，每个部分都有其具体的定义、相关概念和实际应用，对于学习和理解数据结构非常重要。

第四部分 算法

<https://mp.weixin.qq.com/s/auYf1gDpYjKdGozjzGQQ>

算法是信奥的精华部分，也是最难的部分。算法的学习方法是：手工推演+代码练习。算法的大纲要求中的内容，需要理解、掌握，不需要记忆。

大纲要求

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

8.动态规划

[4]动态规划的基本思路

[4]简单一维动态规划

[51 简单背包类型动态规划

[5 简单区间类型动态规划

简要讲解

1. 算法概念与描述：

- 算法概念：解决问题的一系列步骤或指令。
- 算法描述：包括自然语言描述、流程图描述和伪代码描述。自然语言描述使用自然语言来解释算法的步骤；流程图描述使用图形化的方式展示算法的执行流程；伪代码描述是一种类似于编程语言的描述方式，更接近实际代码。

2. 入门算法：

- 枚举法：通过枚举所有可能的解来解决问题，然后从中找到符合条件的解。
- 模拟法：通过模拟问题的实际情况来解决问题，逐步推演出问题的解。

3. 基础算法：

- 贪心法：每一步都选择当前状态下最优的选择，以期望最终得到全局最优解。
- 递推法：通过已知的初始条件和递推关系，逐步推导出问题的解。
- 递归法：将问题分解为更小的子问题，通过递归调用自身来解决问题。
- 二分法：通过将问题的搜索范围逐步缩小一半，来快速定位问题的解。
- 倍增法：通过不断倍增步长，来加速求解问题。

4. 数值处理算法：

- 高精度的加法：用于对大整数进行精确的加法运算。
- 高精度的减法：用于对大整数进行精确的减法运算。
- 高精度的乘法：用于对大整数进行精确的乘法运算。
- 高精度整数除以单精度整数的商和余数：用于对大整数进行除法运算，得到商和余数。

（要会判断在什么时候需要使用高精度！）

5. 排序算法：

- 排序的基本概念：按照某种规则将一组数据按照升序或降序排列。
- 冒泡排序：通过多次比较和交换相邻元素的位置，将最大或最小的元素逐渐移动到最后。
- 选择排序：每次从未排序的数据中选择最小或最大的元素，放到已排序序列的末尾。
- 插入排序：将未排序的数据插入到已排序序列中的适当位置，使得插入后仍然有序。
- 计数排序：通过统计每个元素出现的次数，然后按照元素的大小顺序重新排列。

6. 搜索算法：

- 深度优先搜索：从起始节点开始，沿着一条路径一直向下搜索直到无法继续或者不需要再继续，然后回溯到前一节点继续搜索。
- 广度优先搜索：从起始节点开始，逐层扩展搜索，先搜索距离起始节点最近的节点。

7. 图论算法：

- 深度优先遍历：从图的某个节点开始，沿着一条路径一直深入到无法再继续深入的节点，然后回溯到前一节点继续遍历。
- 广度优先遍历：从图的某个节点开始，逐层扩展遍历，先遍历距离起始节点最近的节点。
- 泛洪算法：从某个节点开始，通过不断扩展泛洪区域，将相同属性的节点标记为同一类。

8. 动态规划：

- 动态规划的基本思路：将复杂的问题分解为简单的子问题，并通过保存子问题的解来避免重复计算，最终得到问题的解。
- 简单一维动态规划：通过一维数组来保存子问题的解，通过迭代求解最优解。
- 简单背包类型动态规划：通过背包问题的状态转移方程，动态地求解背包问题的最优解。

- 简单区间类型动态规划：通过区间问题的状态转移方程，动态地求解区间问题的最优解。

第五部分 数学

https://mp.weixin.qq.com/s/eNGXIP9qwSL8cRvMpbO_jA

数学相关

1. 数及其运算

自然数、整数、有理数、实数及其算术运算 (加、减、乘、除)

- 自然数：自然数是指 1 开始的正整数序列 $\{1, 2, 3, \dots\}$ 。自然数是人类最早使用的数字，也是最基本的数据类型。
- 整数：整数是指全体自然数和它们的相反数，即 $\{\dots -3, -2, -1, 0, 1, 2, 3, \dots\}$ 。整数可以表示各种各样的数量，如人口、面积、温度等。
- 有理数：有理数是指可以表示成两个整数的比值的形式 a/b 的数，其中 a 和 b 是整数，且 b 不为 0。有理数包括自然数、整数、分数和小数。
- 实数：实数是指可以表示成无穷小数的数。实数包括有理数和无理数。无理数是指不能表示成两个整数的比值的形式的数，例如圆周率 π 。

算术运算：

- 加法：加法是指两个数相加得到一个新数的过程。加法的符号是“+”。
- 减法：减法是指一个数减去另一个数得到一个新数的过程。减法的符号是“-”。
- 乘法：乘法是指两个数相乘得到一个新数的过程。乘法的符号是“ \times ”或“ \cdot ”。
- 除法：除法是指一个数除以另一个数得到一个新数的过程。除法的符号是“ \div ”或“/”。

2. 初等数学

代数（初中部分）：

- 一元一次方程：一元一次方程是指只有一个未知数且未知数的指数为 1 的方程。例如： $3x + 5 = 11$ 。
- 一元二次方程：一元二次方程是指只有一个未知数且未知数的指数为 2 的方程。例如： $x^2 + 2x - 3 = 0$ 。

- 因式分解：因式分解是指将一个多项式分解成几个因式的过程。例如： $x^2 + 2x - 3 = (x + 3)(x - 1)$ 。
- 二次根式：二次根式是指含有平方根的表达式。例如： $\sqrt{2}$ 、 $\sqrt{x + 1}$ 。

几何（初中部分）：

- 平面几何：平面几何是指在平面上研究各种几何图形的性质的几何学分支。例如：点、线、角、三角形、四边形等。
- 立体几何：立体几何是指在三维空间中研究各种几何图形的性质的几何学分支。例如：长方体、正方体、球体等。
- 相似形：相似形是指形状和大小相同的图形。例如：两个正方形相似，两个正方体相似。
- 勾股定理：勾股定理是毕达哥拉斯定理的简称，它是指直角三角形中两条直角边平方和等于斜边平方的定理。

3. 初等数论

整除、因数、倍数、指数、质（素）数、合数：

- 整除：整除是指一个整数可以被另一个整数整除，即没有余数。例如：12 整除 6，因为 $12 \div 6 = 2$ ，余数为 0。
- 因数：因数是指一个整数可以整除另一个整数的整数。例如：2、3、4、6 都是 12 的因数。
- 倍数：倍数是指一个整数可以被另一个整数整除的整数。例如：6、12、18 都是 3 的倍数。
- 指数：指数是指一个数的幂次。例如： $2^3 = 8$ ，其中 3 是 2 的指数。
- 质（素）数：质数是指只能被 1 和它本身整除的正整数。例如：2、3、5、7、11 是质数。
- 合数：合数是指可以被 1 和它本身之外的其他正整数整除的正整数。例如：4、6、8、9、10 是合数。

取整：

- 向下取整：向下取整是指将一个小数的尾数部分舍去，得到一个不大于小数的整数。例如：向下取整 3.1415926 得到 3。
- 向上取整：向上取整是指将一个小数的尾数部分舍去，得到一个不小于小数的整数。例如：向上取整 3.1415926 得到 4。
- 四舍五入：四舍五入是指将一个小数的尾数部分舍去，得到一个最接近小数的整数。例如：四舍五入 3.1415926 得到 3。

模运算与同余：

- 模运算：模运算是指将两个整数相除，然后取余数的过程。例如： $7 \% 3 = 1$ ，因为 $7 \div 3 = 2$ ，余数为 1。
- 同余：同余是指两个整数对某个模数取余数后相等。例如： $7 \equiv 1 \pmod{3}$ ，因为 $7 \% 3 = 1$ ， $1 \% 3 = 1$ 。

整数唯一分解定理：

- 整数唯一分解定理：整数唯一分解定理是指每个大于 1 的整数都可以唯一地分解成质数的乘积。例如： $12 = 2^2 \times 3$ 。

辗转相除法（欧几里得算法）：

- 辗转相除法（欧几里得算法）：辗转相除法（欧几里得算法）是一种求两个整数的最大公约数的算法。辗转相除法的基本思想是：两个整数的最大公约数等于较小整数与较大整数相除的余数与较大整数的最大公约数。例如：求 12 和 18 的最大公约数，可以先计算 $18 \div 12 = 1$ ，余数为 6。然后计算 $12 \div 6 = 2$ ，余数为 0。因此，12 和 18 的最大公约数为 6。

4. 素数筛法：埃氏筛法与线性筛法

埃氏筛法和线性筛法都是求素数的算法。它们的基本思想都是将所有大于 2 的偶数筛除，然后将剩下的奇数依次标记为素数，然后……

埃氏筛法：

埃氏筛法的步骤如下：

1. 将所有大于 2 的偶数筛除。
2. 从 3 开始，筛除之后所有 3 的倍数。
3. 从 5 开始，将 5 之后所有 5 的倍数筛除。
4. 从下一个奇数开始，直到筛除到最大的奇数。

线性筛法：

线性筛法的步骤如下：

1. 将部分大于 2 的偶数筛除。

2. 从 3 开始，将部分 3 的倍数筛除。
3. 从 5 开始，将部分 5 的倍数筛除。
4. 从下一个标记的素数开始，直到筛除到最大的数；后续筛选的时候，适量筛除之前未筛除的倍数。

与埃氏筛法相比，线性筛法在标记素数时更加高效。这是因为线性筛法只标记每个合数的一次，而埃氏筛法需要标记所有倍数，会有重复筛除。

4. 离散与组合数学

集合：

集合是指由一组元素组成的数学对象。例如： $\{1, 2, 3\}$ 是一个集合，其中 1、2、3 是集合的元素。

集合的基本运算：

- **并集 \cup ：**并集是指两个集合中所有元素组成的集合。例如： $\{1, 2\}$ 和 $\{2, 3\}$ 的并集是 $\{1, 2, 3\}$ 。
- **交集 \cap ：**交集是指两个集合中所有公共元素组成的集合。例如： $\{1, 2\}$ 和 $\{2, 3\}$ 的交集是 $\{2\}$ 。
- **补集（符号为集合上面加一横）：**一个集合补集是指全集中所有未在这个集合中出现的元素组成的集合。例如： $\{1, 2, 3\}$ 相对于全集 $\{1, 2, 3, 4, 5\}$ 的补集是 $\{4, 5\}$ 。

加法原理：

加法原理是指如果一个事件可以分成两类，第一类中的事件可以有 m 种不同的发生方式，另一类中的事件可以有 n 种不同的发生方式，那么这整个事件的发生方式有 $m + n$ 种。

乘法原理：

乘法原理是指如果一个事件可以分成两步，第一步可以有 m 种不同的发生方式，第二步有 n 种不同的发生方式，那么这个事件的发生方式有 $m \times n$ 种。

排列：

排列是指从一组元素中取出一定数量的元素并按一定顺序排列的组合。例如：从集合 {1, 2, 3} 中取出 2 个元素并按一定顺序排列，可以得到 6 种不同的排列：{1, 2}、{1, 3}、{2, 1}、{2, 3}、{3, 1}、{3, 2}。

组合：

组合是指从一组元素中取出一定数量的元素但不考虑顺序的组合。例如：从集合 {1, 2, 3} 中取出 2 个元素但不考虑顺序，可以得到 3 种不同的组合：{1, 2}、{1, 3}、{2, 3}。

杨辉三角：

杨辉三角是一个以数字组成的三角形，其中每行的数字是上一行两个相邻数字的和。杨辉三角的第一个数字是 1，第二行是 1 和 1，第三行是 1、2 和 1，依此类推。杨辉三角有许多有趣的性质，例如：每一行的数字之和等于 2 的幂。

5. 其他

ASCII 码-美国信息交换标准编码：

ASCII 码是 American Standard Code for Information Interchange 的缩写，是一种字符编码方案。ASCII 码使用 7 位二进制位来表示每个字符，因此可以表示 128 个不同的字符。ASCII 码包括英文字母、数字、标点符号和一些特殊字符。

格雷码：

格雷码是一种二进制编码方案，其中相邻两个数字的二进制位只有 1 位不同。编码方式：

1 位的格雷码：0, 1；

2 位的格雷码：前两个数在 1 位的基础上前面加 0，后两个数把 1 位的编码顺序倒过来，再在前面加 1；

n 位：0+ (n-1) 位；1+ (n-1) 位的倒序。