**CSP-J/S 第一轮知识点选讲**

**信息学史及基本知识**

**一、信息学及计算机史**

**计算机的顶级奖项**：图灵奖、冯·诺依曼奖

图灵奖：由ACM（美国计算机协会）设立于1966年。是“计算机界的诺贝尔奖”。

冯·诺依曼奖：由IEEE设立。

**对信息科学做出突出贡献的大神**：图灵（所以才有个奖），冯·诺伊曼

**中国获图灵奖的大神**：姚期智（清华就有姚班，就是以他的名字命名的）

**世界第一台电子计算机**：埃尼阿克（ENIAC），于1946年2月14日在美国宾夕法尼亚大学诞生。又被叫做电子管计算机。

**二、关于编程**

**编程语言**：

分两类：面向对象和面向过程。

**高级语言和低级语言的区别**：

高级语言需要编译运行，常数较大，运行速度慢。而低级语言常数极小，运行速度快。此外，高级语言更容易移植。

**常见低级语言**：

汇编

**面向对象的高级语言**：

C++，Java，EIFFEL，Simula 67等。

**面向过程的高级语言**：

C，Fortran语言。

**递归编程**：

递归是指一种通过重复将问题分解为同类的子问题而解决问题的方法。递归式方法可以被用于解决很多的计算机科学问题。简单来讲，就是“自身调用自身”（在函数中）。

**P类/NP类/NPC类问题**：

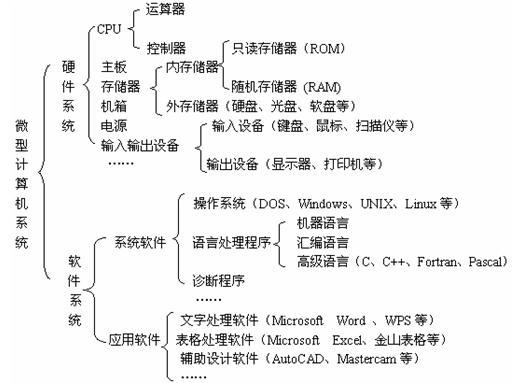
1、P类问题：如果一个问题能找到一个在多项式时间内解决它的算法，那么这个问题就是P问题。

2、NP类问题：注意：NP问题**不是非P类问题**，而是在多项式时间内验证一个解的问题。或者，我们可以将其理解为在多项式时间内猜出一个解的问题。

3、NPC类问题：定义如下：如果一个问题是NP问题，而且所有的NP问题都可以约化到它。那么它就是NPC类问题。再来介绍一下关于约化的定义：如果一个问题A可以约化为问题B，含义就是这个问题A可以用问题B的解法来解决。

**三、关于计算机**

先上张大图：



**重要设备**：

硬件组成：

控制器(Control):是整个计算机的中枢神经，其功能是对程序规定的控制信息进行解释，根据其要求进行控制，调度程序、数据、地址，协调计算机各部分工作及内存与外设的访问等。

运算器(Datapath):运算器的功能是对数据进行各种算术运算和逻辑运算，即对数据进行加工处理。

存储器(Memory):存储器的功能是存储程序、数据和各种信号、命令等信息，并在需要时提供这些信息。

输入设备(Input system):输入设备是计算机的重要组成部分，输入设备与输出设备合称为外部设备，简称外设，输入设备的作用是将程序、原始数据、文字、字符、控制命令或现场采集的数据等信息输入到计算机。常见的输入设备有键盘、鼠标器、光电输入机、磁带机、磁盘机、光盘机等。

输出设备(Output system):输出设备与输入设备同样是计算机的重要组成部分，它把外算机的中间结果或最后结果、机内的各种数据符号及文字或各种控制信号等信息输出出来。微机常用的输出设备有显示终端CRT、打印机、激光印字机、绘图仪及磁带、光盘机等。

**CPU及存储**：

CPU（中央处理器）=运算器+控制器+寄存器

运算器=算术逻辑运算单元（ALU）及浮点运算单元（FPU）

存储器=内存储器+外存储器

BIOS是英文"Basic Input Output System"的缩略语，直译过来后中文名称就是"基本输入输出系统"。其实，它是一组固化到计算机内主板上一个ROM芯片上的程序，它保存着计算机最重要的基本输入输出的程序、系统设置信息、开机后自检程序和系统自启动程序。 其主要功能是为计算机提供最底层的、最直接的硬件设置和控制。

随机存储器RAM的“随机”指“随时访问”

所以，我们记下来以下知识点：

**断电后可以保存数据：硬盘，ROM**

**断电后不可以保存数据：显存（显卡内存），RAM，CPU**

**计算机各存储单位及进位关系**：

计算机的存储单位有以下几种：

TB/GB/MB/KB/B

他们之间的进位关系为1024（常识）

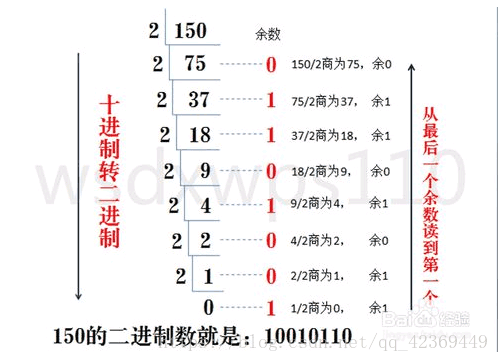
特殊地，1B=8(bit)，这里的bit是二进制下的一位内存。

**进制及进制转化**

**十进制转任意进制**

将十进制转换成N进制，只需把十进制数每次除N求余数，然后把余数逆序写出来。

看不懂就看图：

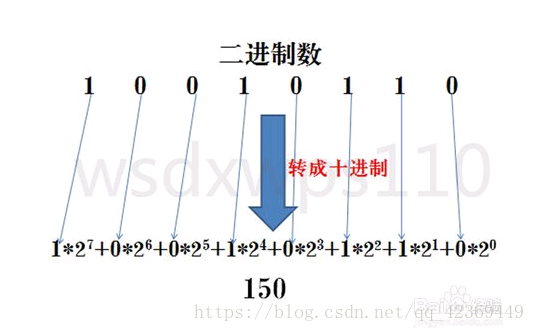


这是二进制的图，其他进制就类比推一下就可以了。

**任意进制转十进制**

简单说就是：按位转，第i位的数字乘以要转换的进制的n−1次幂即可。

还是上图：



**任意进制互相转化**

这里考虑用十进制做中转，先把A进制转十进制，再把十进制转B进制。

**关于小数的进制转换**

十进制转任意进制的小数不进行除法运算，而进行乘法运算后取整，取整后从前向后排列。

任意进制转十进制的小数只需要乘上负指数，最后算出来即可。

**各进制的字母表达**

H(Hexadecimal)——16进制

D(Decimal)——10进制

O(Octonary)——8进制

B(Binary)——2进制

**二进制的相关知识**

二进制是计算机进行计算所使用的工具，自然也是非常常考的要点。二进制的相关知识有许多，甚至算法中的位运算也是二进制的相关内容，但为了过第一轮初赛，我们只介绍一些理论知识。关于位运算的相关知识请有兴趣的同学自己学习。

**1、原码**

顾名思义，原码就是十进制数直接转换成二进制之后直接形成的二进制编码。

**2、补码**

正数的补码是本身，负数的补码是其**反码加一**。

**3、反码**

顾名思义：正数的反码是本身，负数的反码是其除符号位之外的所有位按位取反的结果。

**附：ASCII码**

ASCII码的正规名称是：**美国信息交换标准代码**，是基于拉丁字母的一套电脑编码系统。是最通用的信息交换标准。一共定义了**128**个字符。

这里不赋ASCII码的转换表。只给出几种比较常用的转换：

字符0→48

大写字母A→65

小写字母a→97

空格→32

换行→13

**逻辑运算**

**逻辑运算**

逻辑运算一共有三种，每种都有两种写法：

**逻辑非**：！或 ┐

**逻辑与**：&& 或 ∧

**逻辑或**：|| 或 ∨

**逻辑运算的优先级**

**非>与>或**

**位运算+逻辑运算的优先级**

**逻辑非（！，┐）= 按位反（~）> 位移运算（<<,>>）> 不等号（>=,<=）> 等号（==,!=）> 按位与（&）> 按位异或（^）> 按位或（|）> 逻辑与（&&，∧）> 逻辑或（||，∨）**

**逻辑表达式**

由逻辑运算复合而成，只有两种结果：true和false，在C/C++中，返回的值以0表示假，以1表示真。

**条件表达式**

条件表达式的基本形式如下：

<表达式1>？<表达式2>：<表达式3>

其表达意义是：如果表达式1成立，则执行表达式2，否则执行表达式3。其实也等价于if−else条件语句。例如下：

#define Min(a,b) a<b?a:b

**注意：**如果条件表达式有多个进行复合，那么在执行的时候需要从由往左依次判断最后得出一个结果。即：**右结合性**。

比如：

<表达式1>？<表达式2>：<表达式3>？<表达式4>：<表达式5>

那么，在执行的时候是从3开始判断是否为真，然后执行某一个表达式，依次向上回溯。

**图论理论知识**

**基本概念**

**完全图**：任意两点都有边相连，我们很容易推出来，一张完全图的边数为（n为节点个数）

n×(n−1)2

**连通图**：顾名思义，连通图就是连通的图，即任意两点都能**直接或间接**到达，这就区别于完全图必须直接用边到达的定义。

**树**：直观来讲，就是一张长得像树的图。定义是任意两点之间的简单路径有且只有一条。树是一棵连通且无环的图。它的边数是n−1。

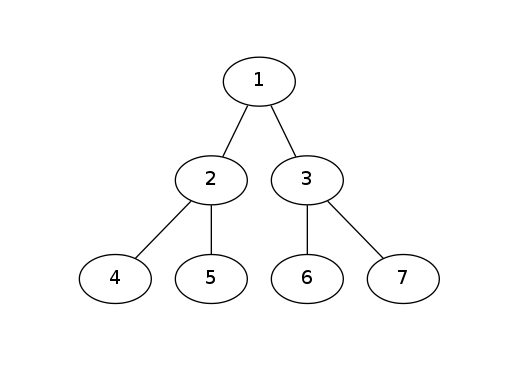
**二叉树的遍历**

二叉树有不同的遍历方式，一般来讲，我们将其分成三类：先序遍历（也叫先根遍历）、中序遍历（中根遍历）以及后序遍历（后根遍历）。

**先序遍历**：遍历方式如下：根—左儿子—右儿子

**中序遍历**：遍历方式如下：左儿子—根—右儿子

**后序遍历**：遍历方式如下：左儿子—右儿子—根

我们用一张图来理解一下这几种遍历方式。

**这张图的先序遍历：1245367**

**中序遍历：4251637**

**后序遍历：4526731**

**一个推论**：

先序遍历+中序遍历=一棵确定的二叉树

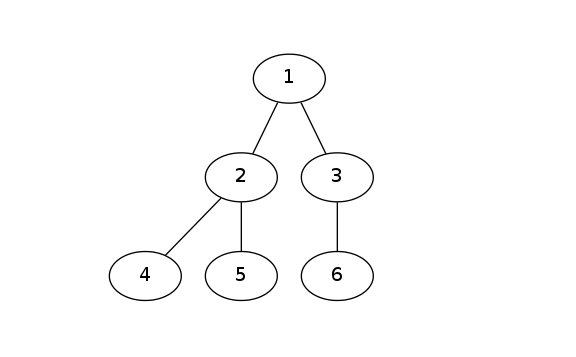
后序遍历+中序遍历=一棵确定的二叉树

先序遍历+后序遍历=啥也不是

**特殊二叉树及其性质**

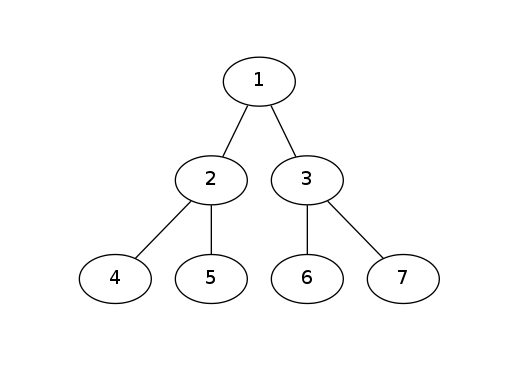
**完全二叉树**：只有最后一层不是满的，且最后一层的所有节点均集中在左侧。

图例如下：



**满二叉树**：节点个数已满。

图例如下：



**特殊二叉树的性质**：

1、对于一棵完全二叉树来讲，它的叶子节点为n，则节点总数为2×n−1。此结论可逆。

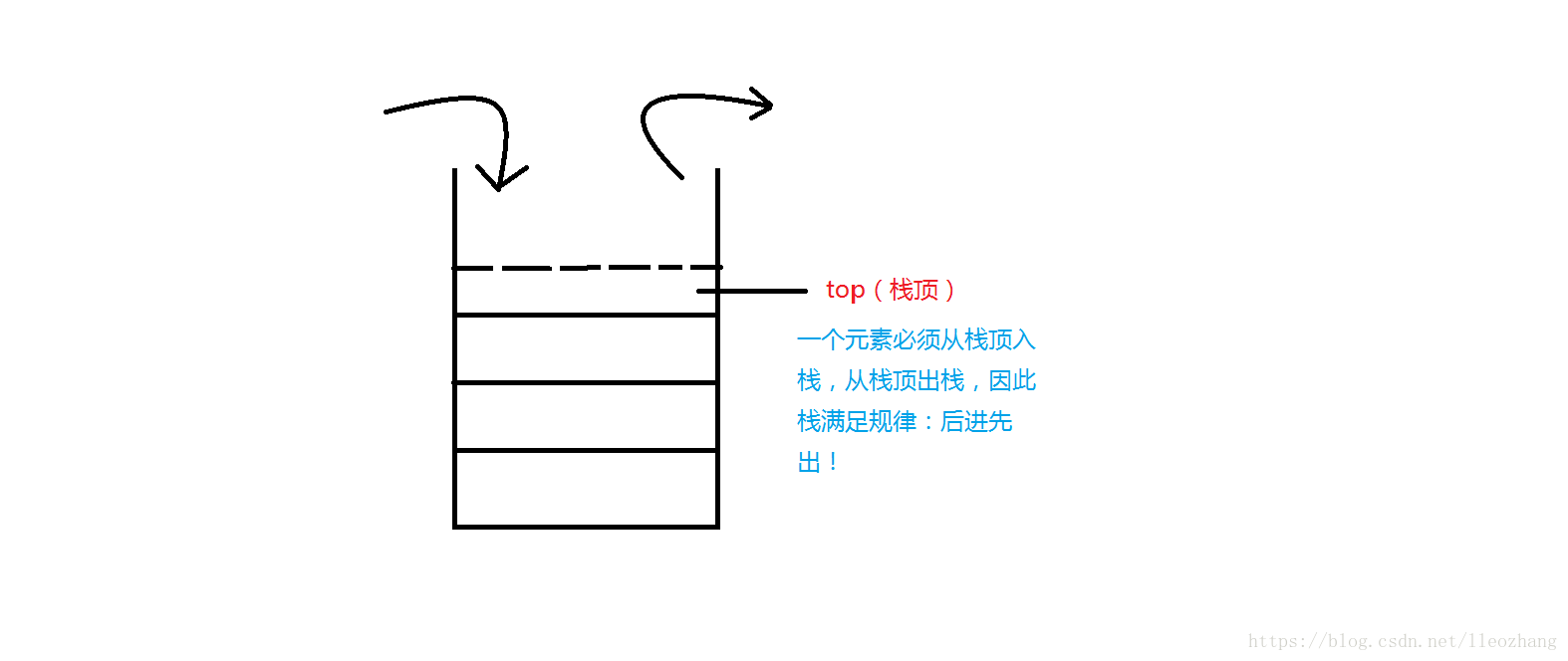
2、对于一棵满二叉树来讲，它的层数（深度）为k，则它的节点总数为2k−1。此结论可逆。

**简单数据结构基本理论**

**1、栈**

想象一个桶，你从上面往里扔砖，然后你想把某一块砖拿出来，你需要先拿出来你后扔进去的砖。这就是栈。栈的基本原则是：后进先出

来一发图示？



**附：前、中、后缀表达式**

**中缀表达式：**6×12+9−14

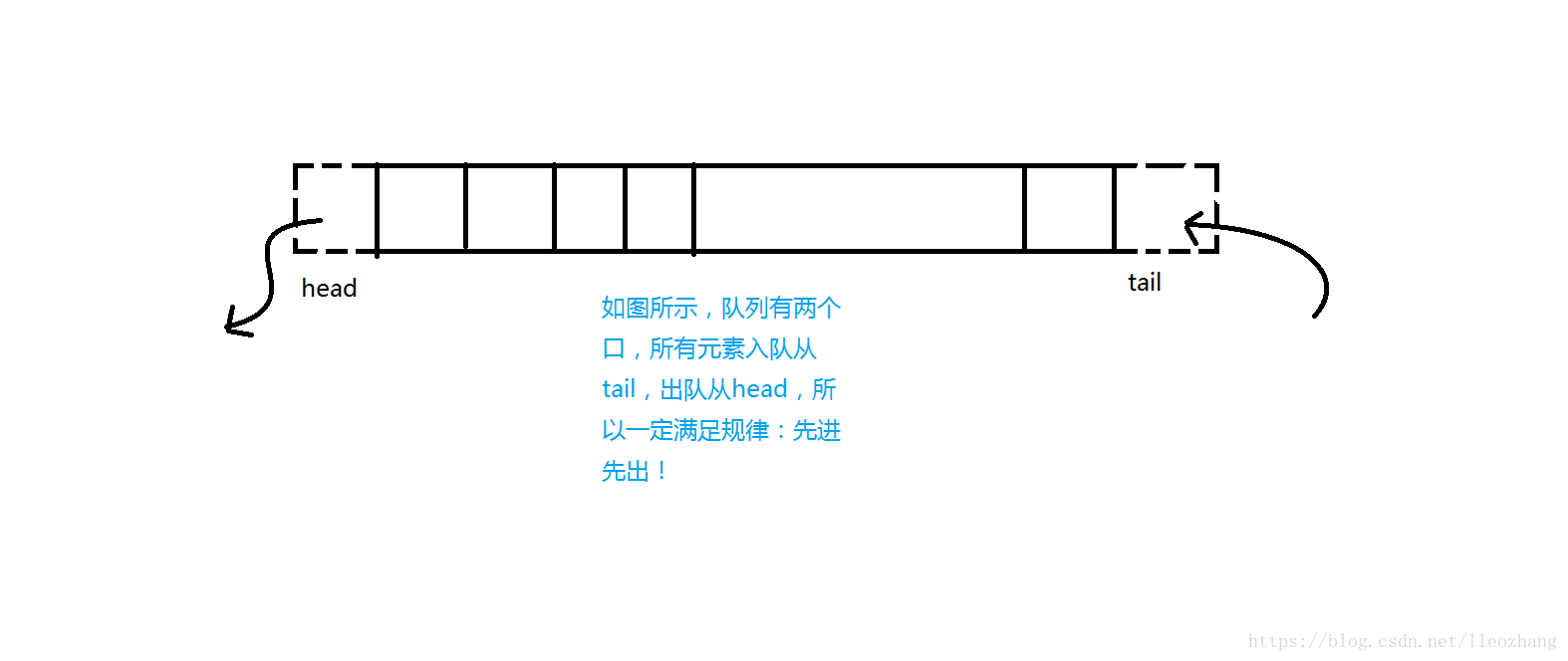
**后缀表达式：首先，建一个存运算符号的栈，一个存数字的栈。然后，碰到一个符号，就压到符号栈中，碰到数字就压到数字栈中，如果数字栈中有两个数，就从符号栈中弹出一个符号来对这两个数进行运算，把结果压回数字栈中，以此类推。**×6+12 9−14

**前缀表达式：14−9 12+6×**

**2、队列**

想象你在排队买票，这个队伍中的人都非常有素质，都自觉排队而且不会提前离开队伍。这样就只能从队首买完票再离开，从队尾进入队伍。队列的基本原则是：先进先出。

再来一发图示：



**3、链表**

链表分两种：单向链表和双向链表。

**4、字符串**

字符串子串的概念：字符串是一串字符~~（废话）~~，它的子串被定义为：字符串中**任意个连续的字符组成的子序列**。

字符串子串个数的计算公式：

n×(n+1)2+1

（就是字符串长度等差数列）

如果是非空子串，就把那个一减去即可（子串个数的公式加一就是考虑空子串的情况）。

**时空复杂度的计算**

**时间复杂度**：渐进时间复杂度用符号O 表示。一个程序的语句执行次数可以用一个代数式表示，那么我们取这个代数式的最高次项且忽略此项系数作为时间复杂度。如果一个程序的语句执行次数为 2n3+3n2+n+7，那么这个程序的渐进时间复杂度为O(n3)。

**计算非递归程序的时间复杂度**：简单粗暴，数循环。

**常数**：常数即为我们忽略掉的O中最高次项的系数与低次项所带来的时间消耗。

**空间复杂度**：类比时间复杂度。看开空间开了多大。

**计算空间占用量**：根据我们以上说过的计算机存储单位的知识：一个int占用的内存是4B，所以我们把开的int乘上4，再除以1024就是KB，同理，再除1024就是MB。

公式：n为元素个数，M为最终答案（以MB为单位）

M=4n1024×1024

PS:一般来讲，比赛中所给的256MB内存可以开6×107个int类型的变量。另外，大数组必须开全局变量。如果扔在主函数里极容易爆栈。

**数学、逻辑学及运筹学知识**

**排列组合**：排列组合是每年必考知识点。但是这是一个比较大的课题。不仅是高二数学选修重点，也是数学编程的一个重要分支。

1、排列和组合的定义

(1)排列的定义

从n个不同元素中，选出m个元素按照一定顺序排成一列，叫做从n个不同元素中取出m个元素的一个排列。

(2)排列数的定义

从n个元素中选出m个元素的所有排列的个数，叫做从n个不同元素中取出m个元素的排列数。

(3)全排列的定义

当n=m时所有的排列情况叫做全排列。

(4)组合的定义

从n个不同元素中，选出m个元素并成一组，叫做从n个不同元素中取出m个元素的一个组合。

(5)组合数的定义

从n个元素中选出m个元素的所有组合的个数，叫做从n个不同元素中取出m个元素的组合数。

(6)排列&组合的区别

通俗地说，组合不分顺序，而排列分顺序，也就是说，对于数列1,2，有以下两种排列：1,2和2,1，但是仅有一种组合1,2或2，1.

2、排列&组合的公式

(1)关于排列的公式

从n个不同元素中，选出m个元素的排列数，数学表示为：Amn.

计算公式如下：

Amn=n(n−1)(n−2)⋯(n−m+1)=n!(n−m)!

(2)关于组合的公式

从n个不同元素中，选出m个元素的组合数，数学表示为：Cmn.

计算公式如下：

Cmn=Amnm!=n!m!(n−m)!

(3)关于全排列的公式

某个数列的全排列数f(n)，计算公式如下：

f(n)=n!

**题目类型整理**

| **题型** | **知识点类型** | **题目数量** |
| --- | --- | --- |
| 单选 | 信息学史&基本知识 | 8-10 |
| 单选 | C++语法知识点 | 2-3 |
| 单选 | 数据结构&算法 | 3-4 |
| 单选 | 数学&逻辑学&运筹学 | 3-4 |
| 单选 | 比赛相关知识 | 1-2 |
| 问题求解 | 数学 | 1 |
| 问题求解 | 数据结构 | 1 |
| 模拟程序运行 | C++语法&算法 | 4 |
| 完善程序 | C++语法&算法 | 2 |