

# CSP 初赛典型题目复习

(第一届 CCF CSP 比赛于 2019 年举行)

## 一、基础知识

### 1、稳定排序与不稳定排序归纳

稳定排序：

如：插入排序，基数排序，归并排序，冒泡排序，计数排序。

不稳定排序

不稳定的排序算法有：快速排序（相同会交换），希尔排序，简单选择排序，堆排序

### 2、NOI 序列比赛中，选手提交的程序源文件必须不大于 100KB。

### 3、【CSP2020-S】表达式 $a*(b+c)-d$ 的后缀表达形式为（ ）。

- A.  $abc*d-$       B.  $-+*abcd$       C.  $abcd*+-$       D.  $abc+*d-$

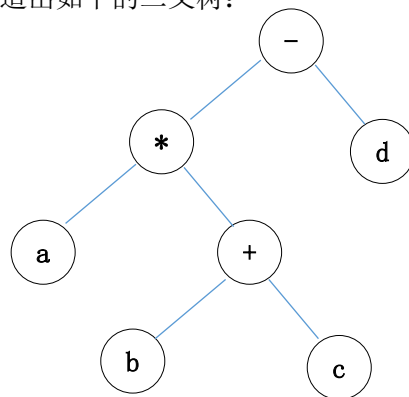
分析：

中缀表达式转为后缀表达式，或者中缀表达式转为前缀表达式的方法：可以通过画出中缀表达式对应的二叉树来实现。

画树的原则：数字只能做叶子节点，运算符作父节点，左操作数作为左子树，右操作数作为右子树，从最先被计算的那个算式开始画。

树构造出来后，根据前序遍历，得出前缀表达式；根据后序遍历，得到后缀表达式。

比如样例可以构造出如下的二叉树：



该二叉树的后序遍历为： $abc+*d-$ ，即为后缀表达式。答案选（ D ）

4、【NOIP2005】一位艺术史学家有 20000 幅  $1024 * 768$  的真彩色图像（32 位），如果将这些图像以位图形式保存在 CD 光盘上（一张 CD 光盘的容量按 600M 计算），大约需要（ ）张 CD 光盘。

- A. 1                      B. 10                      C. 100                      D. 1000                      E. 10000

分析： $(1024 * 768 * 32 * 20000) / (8 * 1024 * 1024 * 600) = 100$  张。答案选（ C ）

5、【NOIP2005 提高组】处理器 A 每秒处理的指令数是处理器 B 的 2 倍。某一特定程序 P 分别编译为处理器 A 和处理器 B 的指令，编译结果处理器 A 的指令数是处理器 B 的 4 倍。已知程序 P 的算法时间复杂度为  $O(n^2)$ ，如果处理器 A 执行程序 P 时能在一小时内完成的输入规模为  $n$ ，则处理器 B 执行程序 P 时能在一小时内完成的输入规模为（ ）。

- A.  $4 * n$                   B.  $2 * n$                   C.  $n$                       D.  $n/2$                   E.  $n/4$

分析：每秒指令数 2:1，编译后指令量 4:1，综合速度 1:2，无论在哪一个处理器上，该程序 P 的算法复杂度都是一样的，因此，产生影响的主要是指令数及两个处理器的速度。综合起来看，处理器 B 有两倍的速度优势，因此，在同样的一小时内，B 可以处理多一倍的数据，即输入数据的规模可以是  $2 * n$ 。答案为（ B ）

## 6. 进制转换

### （1）八进制与二进制的转换（1 位八进制数对应 3 位二进制数）

例：将八进制的 37.416 转换成二进制数：

$$\begin{array}{ccccccc} 3 & 7 & . & 4 & 1 & 6 \\ 011 & 111 & . & 100 & 001 & 110 \\ \text{即: } (37.416)_8 & = & (11111.1000111)_2 \end{array}$$

例：将二进制的 10110.0011 转换成八进制：

$$\begin{array}{cccc} 010 & 110 & . & 001 \\ \hline 2 & 6 & . & 1 \\ \text{即: } (10110.011)_2 & = & (26.14)_8 \end{array}$$

### （2）十六进制与二进制的转换（1 位十六进制数对应 4 位二进制数）

例：将十六进制数 5DF.9 转换成二进制：

$$\begin{array}{ccccccc} 5 & D & F & . & 9 \\ 0101 & 1101 & 1111 & . & 1001 \\ \text{即: } (5DF.9)_{16} & = & (10111011111.1001)_2 \end{array}$$

例：将二进制数 1100001.111 转换成十六进制：

$$\begin{array}{ccccccc} 0110 & 0001 & . & 1110 \\ 6 & 1 & . & E \\ \text{即: } (1100001.111)_2 & = & (61.E)_{16} \end{array}$$

7、【NOIP1999】在 24\*24 点阵的字库中，汉字“一”与“编”的字模占用字节数分别是（ ）。

- A. 32、32      B. 32、72      C. 72、72      D. 72、32

分析：24×24 点阵的字模需要 24×24 个二进制位来存储，每个字节有 8 个二进制位。所以一个汉字字模占用字节。答案为（ C ）。

8、【NOIP1999】组成“教授”（JIAO SHOU），“副教授”（FU JIAO SHOU）与“讲师”（JIANG SHI）这三个词的汉字，在 GB2312-80 字符集中都是一级汉字，对这三个词排序的结果是（ ）。

- A. 教授、副教授、讲师      B. 副教授、教授、讲师  
C. 讲师、副教授、教授      D. 副教授、讲师、教授

分析：GB2312-80 是我国于 1981 年颁布的《信息交换用汉字编码字符集》，字符集中共收录 6763 个汉字，其中一级字库 3755 个，按拼音排序，二级字库 3008 个，按偏旁部首排序，另外还有 682 个图文符号。因副教授，讲师，教授三个词的汉字都在一级字库，所以按拼音排序是副教授、讲师、教授。答案为（ D ）。

9、【NOIP2001 提高组】64KB 的存储器用十六进制表示，它的最大的地址码是（ ）。

- A. 10000      B. FFFF      C. 1FFFF      D. EFFFF

分析：64KB 即  $64 \times 1024$  字节，由  $64=2^6$ 、 $1024=2^{10}$  得  $64 \times 1024=2^{16}$

要表示  $2^{16}$  个存储单元，需 16 根地址线，最大地址码为 1111111111111111，用十六进制表示 FFFF。答案为（ B ）。

10. 【NOIP1999】计算机中的数有浮点与定点数两种，用浮点数表示的数，通常由（ ）两部分组成。

- A. 指数与基数      B. 尾数与小数      C. 阶码与尾数      D. 整数与小数

分析：计算机中的数如  $1.701412E+09$ ，叫指数计数法，也叫浮点计数法，其一般形式为  $\pm NE \pm nn$ ， $\pm N$  叫尾数，可以是一位非零整数或整数部分是一位非零数字的小数，小数点后面最多可有 6 位数字。表示正数时，正号省略掉。E 是英文 exponent 的第一个字母的大写形式，表示指数。后面  $\pm nn$  叫阶码，阶码有两整数位，表示尾数所乘 10 的幂的指数。绝对值小于 10 的阶码，输出时十位补 0，阶码为正时+号不省略。答案为（ C ）

11. 【NOIP2001 提高组】 $[x]$  补码=10011000，其原码为（ ）。

- A. 011001111      B. 11101000      C. 11100110      D. 01100101

分析：对于正数来说， $[x]$  原码= $[x]$  反码= $[x]$  补码。对于负数来说， $[x]$  补码= $[x]$  反码+1、 $[x]$  反码等于  $[x]$  原码除符号位外逐位取反。所以  $[x]$  原码等于  $[x]$  补码-1 且除符号位逐位取反，得 11101000。答案为（ B ）。

12. 【NOIP2002 提高组】已知  $x = (0.1011010)_2$ ，则  $[x/2]$  补 = ( )<sub>2</sub>。

- A. 0.1011101      B. 1.1110110      C. 0.0101101      D. 0.1001110

分析：因  $[x/2]$  原码 = 0.0101101， $[x/2]$  补 =  $[x/2]$  原码，所以答案为 C。答案为 ( C )

12. 若  $A = \text{True}$ ， $B = \text{False}$ ， $C = \text{True}$ ， $D = \text{False}$ ，以下逻辑运算表达式真的有 ( )。

- A.  $(A \wedge B) \vee (C \wedge D \vee \neg A)$       B.  $((\neg A \wedge B) \vee C) \wedge \neg B$   
C.  $(B \vee C \vee D) \vee D \wedge A$       D.  $A \wedge (D \vee \neg C) \wedge B$

分析：考查逻辑运算的知识， $\wedge$  相当于与 &&， $\vee$  相当于或 ||， $\neg$  相当于!(取反)。

一个个算结果，比如 A 选项  $(A \wedge B) \vee (C \wedge D \vee \neg A)$ ，根据运算级的比较，我们可以定下运算的顺序，然后按运算顺序计算结果。注意，这类题是有个小技巧的。比如 A 选项可以先看中间的  $\vee$ ，为什么呢？因为  $\vee$  的左右有一边是真就行，可以不去看另外一边。

A 选项的结果是： $(A \wedge B) \vee (C \wedge D \vee \neg A)$ ， $(A \wedge B) = \text{假}$ ， $(C \wedge D \vee \neg A)$  中  $C \wedge D = \text{假}$ ， $\neg A = \text{假}$ ，所以  $(C \wedge D \vee \neg A) = \text{假}$ 。于是 A 选项可以简写为： $\text{假} \vee (\text{假} \vee \text{假}) = \text{假}$ 。

B 选项的结果是： $((\neg A \wedge B) \vee C) \wedge \neg B$ ，如果  $\neg B$  是假那么就可以不去看前面的  $((\neg A \wedge B) \vee C)$ ，可惜的是  $\neg B$  是真，那么就要看  $((\neg A \wedge B) \vee C)$ ，发现 C 是真，所以不看  $(\neg A \wedge B)$ ，于是 B 选项可以简写为： $(? \vee \text{真}) \wedge \text{真} = \text{真}$ 。

C 选项的结果是： $(B \vee C \vee D) \vee D \wedge A$ ， $D \wedge A = \text{假}$ ，所以不得不看前面部分  $(B \vee C \vee D)$ ，只要 BCD 有一个是真，那么  $(B \vee C \vee D) = \text{真}$ ，而容易发现  $C = \text{true}$ 。所以 C 选项可以简写为： $\text{真} \vee \text{假} = \text{真}$ 。

D 选项的结果是： $A \wedge (D \vee \neg C) \wedge B$ ，我们很容易发现 D 选项的特殊结构为  $? \wedge ? \wedge ?$ ，三个? 有一个是假，那么 D 为假，A 和 B 不用计算便可看出，所以先发现  $B = \text{假}$ ，所以  $D = \text{假}$ 。

答案为：BC。