



2020 年 CSP-J 组 第一轮真题讲解

by Ufowoqqqo

2021-08-15



www.luogu.com.cn

一、单项选择题

共 15 题，每题 2 分，共计 30 分；每题有且仅有一个正确选项

第 1 题

在内存存储器中每个存储单元都被赋予一个唯一的序号，称为（）。

- A. 地址
- B. 序号
- C. 下标
- D. 编号

正确答案：A

解析

程序和数据平常存储在**硬盘**等存储器上，不管你开机或关机了，它们都是存在的，不会丢失。硬盘可以存储的东西很多，但其**传输数据的速度较慢**。所以需要运行程序或打开数据时，这些数据必须从硬盘等存储器上先传到另一种**容量小但速度快得多的**存储器，之后才送入**CPU**进行执行处理。这中间的存储器就是**内存**。

无论何种存储器，软盘、硬盘、光盘或者内存，都有**地址**。因为它们要存储数据，所以就必须按一定的单位的数据分配一个地址。有了地址，程序才能找到这些数据。这很好理解，想想你们家为什么要有门牌号即可。

我们编程中的每一行代码，代码中用到的每个数据，都需要在内存上有其映射地址。

迁移演练

(NOIP 2018 普及组初赛第 1 题) 以下哪一种设备属于输出设备

- A. 扫描仪
- B. 键盘
- C. 鼠标
- D. 打印机

正确答案： D

解析：输出设备（Output Device）是计算机硬件系统的终端设备，用于接收计算机数据的输出显示、打印、声音、控制外围设备操作等。也是把各种计算结果数据或信息以数字、字符、图像、声音等形式表现出来。常见的输出设备有显示器、打印机、绘图仪、影像输出系统、语音输出系统、磁记录设备等。

迁移演练

(NOIP 2016 普及组初赛第 4 题) 以下不是CPU 生产厂商的是 () 。

- A. Intel
- B. AMD
- C. Microsoft
- D. IBM

正确答案： C

解析：摩尔定律是**英特尔**创始人之一戈登·摩尔的经验之谈，其核心内容为：集成电路上可以容纳的晶体管数目在大约每经过18个月便会增加一倍。换言之，**处理器的性能每隔两年翻一倍**。

迁移演练

(NOIP 2016 普及组初赛第 5 题) 以下不是存储设备的是 () 。

- A. 光盘
- B. 磁盘
- C. 固态硬盘
- D. 鼠标

正确答案： D

解析： 常见存储设备包括利用电能方式存储信息的设备如： 各式存储器， 如RAM（Random Access Memory， 随机存取存储器， 高速易失）、 ROM（Read-Only Memory， 只读存储器， 稳定） 等； 利用磁能方式存储信息的设备如： 硬盘、 软盘、 磁带、 U盘； 利用光学方式存储信息的设备如： CD或DVD

迁移演练

(NOIP 2015 普及组初赛第 2 题) 在 PC 机中, PENTIUM(奔腾)、酷睿、赛扬等 是指()。

- A. 生产厂家名称
- B. 硬盘的型号
- C. CPU 的型号
- D. 显示器的型号

正确答案: C

解析: Intel 奔腾、酷睿 (Core)、赛扬 (Cell) 等
AMD 锐龙 (Ryzen)、速龙 (Athlon) 等

第 2 题

编译器的主要功能是（ ）。

- A. 将源程序翻译成机器指令代码
- B. 将源程序重新组合
- C. 将低级语言翻译成高级语言
- D. 将一种高级语言翻译成另一种高级语言

正确答案： A

解析

简单讲，编译器就是将“一种语言（通常为高级语言）”翻译为“另一种语言（通常为低级语言）”的程序。一个现代编译器的主要工作流程：源代码 (source code) → 预处理器 (preprocessor) → 编译器 (compiler) → 目标代码 (object code) → 链接器 (Linker) → 可执行程序 (executables)

高级计算机语言便于人编写，阅读交流，维护。机器语言是计算机能直接解读、运行的。编译器将汇编或高级计算机语言源程序 (Source program) 作为输入，翻译成目标语言 (Target language) 机器代码的等价程序。源代码一般为高级语言 (High-level language)，如Pascal、C、C++、Java、汉语编程等或汇编语言，而目标则是机器语言的目标代码 (Object code)，有时也称作机器代码 (Machine code)。

迁移演练

(NOIP 2017 普及组初赛第 6 题) 下列不属于面向对象程序设计语言的是 ()。

- A. C
- B. C++
- C. Java
- D. C#

正确答案： A

解析：一代机器语言，二代汇编语言，三代高级语言。从描述客观系统来看，程序设计语言可以分为**面向过程语言**（以“数据结构+算法”程序设计范式构成的程序设计语言，C / Fortran）和**面向对象语言**（以“对象+消息”程序设计范式构成的程序设计语言，VB）。

迁移演练

(NOIP 2015 普及组初赛第 3 题) 操作系统的作用是()。

- A. 把源程序译成目标程序
- B. 便于进行数据管理
- C. 控制和管理系统资源
- D. 实现硬件之间的连接

正确答案： C

解析：操作系统（operating system，简称OS）是管理计算机硬件与软件资源的计算机程序。操作系统需要处理如**管理与配置内存、决定系统资源供需的优先次序、控制输入设备与输出设备、操作网络与管理文件系统等基本事务**。操作系统也提供一个**让用户与系统交互**的操作界面。

第 3 题

设 $x=\text{true}$, $y=\text{true}$, $z=\text{false}$, 以下逻辑运算表达式值为真的是 ()。

- A. $(y \vee z) \wedge x \wedge z$
- B. $x \wedge (z \vee y) \wedge z$
- C. $(x \wedge y) \wedge z$
- D. $(x \wedge y) \vee (z \vee x)$

正确答案： D

解析

已知 $x=\text{true}$, $y=\text{true}$, $z=\text{false}$, 则

- A. $(y \vee z) \wedge x \wedge z = (\text{true} \vee \text{false}) \wedge \text{true} \wedge \text{false} = \text{true} \wedge \text{true} \wedge \text{false} = \text{false}$
- B. $x \wedge (z \vee y) \wedge z = \text{true} \wedge (\text{false} \vee \text{true}) \wedge \text{false} = \text{true} \wedge \text{true} \wedge \text{false} = \text{false}$
- C. $(x \wedge y) \wedge z = (\text{true} \wedge \text{true}) \wedge \text{false} = \text{true} \wedge \text{false} = \text{false}$
- D. $(x \wedge y) \vee (z \vee x) = (\text{true} \wedge \text{true}) \vee (\text{false} \vee \text{true}) = \text{true} \vee \text{true} = \text{true}$

迁移演练

(CSP 2019 入门组第一轮第 2 题) 二进制数 11 1011 1001 0111 和 01 0110 1110 1011 进行逻辑与运算的结果是 ()。

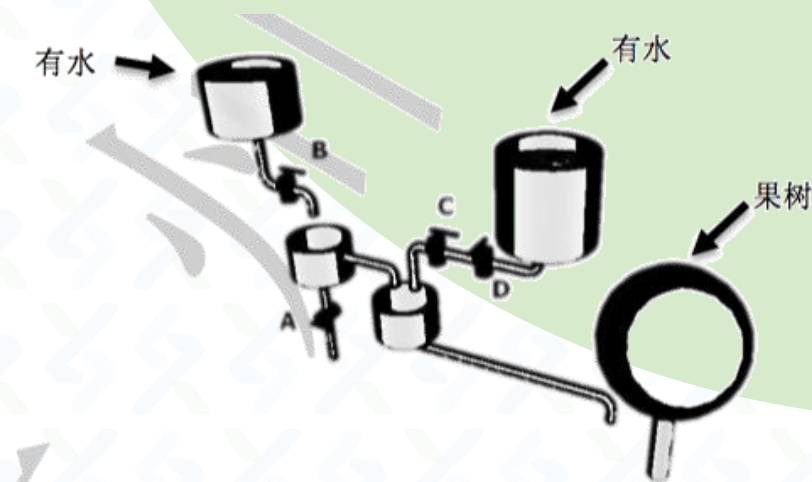
- A. 01 0010 1000 1011
- B. 01 0010 1001 0011
- C. 01 0010 1000 0001
- D. 01 0010 1000 0011

正确答案： D

解析：对于每一位都有 $0 \& 0 = 0$ 、 $0 \& 1 = 0$ 、 $1 \& 0 = 0$ 和 $1 \& 1 = 1$

迁移演练

(NOIP 2016 普及组初赛第 17 题) 右图表示一个果园灌溉系统，有 A、B、C、D 四个阀门，每个阀门可以打开或关上，所有管道粗细相同，以下设置阀门的方法中，可以让果树浇上水的是（ ）。



- A. B 打开，其他都关上
- B. AB 都打开，CD 都关上
- C. A 打开，其他都关上
- D. D 打开，其他都关上

正确答案： A

第 4 题

现有一张分辨率为 2048×1024 像素的 32 位真彩色图像。请问要存储这张图像，需要多大的存储空间？（ ）。

- A. 16MB
- B. 4MB
- C. 8MB
- D. 2MB

正确答案： C

解析

$2048 \times 1024 = 2,097,152$ 像素

$2,097,152 \times 32 = 67,108,864$ 位 (bits)

$67,108,864 \div 8 = 8,388,608$ 字节 (Bytes)

$8,388,608 \div 1,024 = 8,192$ 千字节 (Kilobytes)

$8,192 \div 1,024 = 8$ 兆字节 (Megabytes)

迁移演练

(CSP 2019 入门组第一轮第 3 题) 一个32位整型变量占用 () 个字节。

- A. 32
- B. 128
- C. 4
- D. 8

正确答案： C

解析： $32 \text{ (bits)} \div 8 \text{ (bits per byte)} = 4 \text{ (bytes)}$

迁移演练

(NOIP 2018 普及组初赛第 3 题、NOIP 2015 普及组初赛第 1 题)
1MB 等于 ()

- A. 1000 字节
- B. 1024 字节
- C. 1000 X 1000 字节
- D. 1024 X 1024 字节

正确答案： D

解析： 1 Megabyte = 1,024 Kilobytes

1 Kilobyte = 1,024 Bytes

迁移演练

(NOIP 2017 普及组初赛第 2 题) 计算机存储数据的基本单位是 () 。

- A. Bit
- B. Byte
- C. GB
- D. KB

正确答案： B

解析： 需要注意 bit 虽然是最小单位，但不是基本单位。

迁移演练

(NOIP 2017 普及组初赛第 4 题) 分辨率为 800x600、16 位色的位图，存储图像信息所需的空间为 ()。

- A. 937.5KB
- B. 4218.75KB
- C. 4320KB
- D. 2880KB

正确答案： A

解析： $800 * 600 * 16 \text{ (bits)} / 8 \text{ (bits per byte)} / 1024 \text{ (bytes per kilobyte)} = 937.5 \text{ kilobytes}$

迁移演练

(NOIP 2016 普及组初赛第 2 题) 如果 256 种颜色用二进制编码来表示, 至少需要 () 位。

- A. 6
- B. 7
- C. 8
- D. 9

正确答案: C

解析: $\log_2 256 = 8$

第 5 题

冒泡排序算法的伪代码如下：

对 n 个数用以下冒泡排序算法进行排序，最少需要比较多少次？
()。

- A. n^2
- B. $n-2$
- C. $n-1$
- D. n

输入：数组 L , $n \geq k$ 。输出：按非递减顺序排序的 L 。

算法 BubbleSort:

```
1. FLAG  $\leftarrow$  n // 标记被交换的最后元素位置
2. while FLAG > 1 do
3.   k  $\leftarrow$  FLAG - 1
4.   FLAG  $\leftarrow$  1
5.   for j=1 to k do
6.     if L(j) > L(j+1) then do
7.       L(j)  $\leftrightarrow$  L(j+1)
8.       FLAG  $\leftarrow$  j
```

正确答案： C

解析

时间复杂度

若文件的初始状态是正序的，一趟扫描即可完成排序。所需的关键字比较次数 C 和记录移动次数 M 均达到最小值： $C_{\min} = n - 1$ ， $M_{\min} = 0$ 。

所以，冒泡排序最好的时间复杂度为 $O(n)$ 。

若初始文件是反序的，需要进行 $n - 1$ 趟排序。每趟排序要进行 $n - i$ 次关键字的比较($1 \leq i \leq n - 1$)，且每次比较都必须移动记录三次来达到交换记录位置。在这种情况下，比较和移动次数均达到最大值： $C_{\max} = \frac{n(n-1)}{2} = O(n^2)$

$$M_{\max} = \frac{3n(n-1)}{2} = O(n^2)$$

冒泡排序的最坏时间复杂度为 $O(n^2)$ 。

综上，因此冒泡排序总的平均时间复杂度为 $O(n^2)$ 。

解析

算法稳定性

冒泡排序就是把小的元素往前调或者把大的元素往后调。比较是相邻的两个元素比较，交换也发生在这两个元素之间。所以，如果两个元素相等，是不会再交换的；如果两个相等的元素没有相邻，那么即使通过前面的两两交换把两个相邻起来，这时候也不会交换，所以相同元素的前后顺序并没有改变，所以冒泡排序是一种稳定排序算法。

解析

排序算法	平均时间复杂度	最坏复杂度	空间复杂度	稳定性
冒泡排序	$O(n^2)$	$O(n^2)$	$O(1)$	稳定
选择排序	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
直接插入排序	$O(n^2)$	$O(n^2)$	$O(1)$	稳定
快速排序	$O(n \log n)$	$O(n^2)$	$O(\log n)$	不稳定
归并排序	$O(n \log n)$	$O(n \log n)$	$O(1)$	稳定
堆排序	$O(n \log n)$	$O(n \log n)$	$O(1)$	不稳定
希尔排序	$O(n \log n)$	$O(n^s)$ 1	$O(1)$	不稳定
基数排序	$O(\log_R B)$	$O(\log_R B)$	$O(n)$	稳定
二叉树排序	$O(n \log n)$	$O(n \log n)$	$O(n)$	稳定
计数排序	$O(n + k)$	$O(n + k)$	$O(n + k)$	稳定

迁移演练

(CSP 2019 入门组第一轮第 5 题) 设有 100 个已排序的数据元素，采用折半查找时，最大比较次数为 ()

- A. 7
- B. 10
- C. 6
- D. 8

正确答案： A

解析： $\lceil \log_2 100 \rceil = 7$

迁移演练

(NOIP 2018 普及组初赛第 8 题) 以下排序算法中，不需要进行关键字比较操作的算法是 ()。

- A. 基数排序
- B. 冒泡排序
- C. 堆排序
- D. 直接插入排序

正确答案： A

解析：基数排序（英语：Radix sort）是一种**非比较型**整数排序算法，其原理是将整数按位数切割成不同的数字，然后按每个位数分别比较。由于整数也可以表达字符串（比如名字或日期）和特定格式的浮点数，所以基数排序也不是只能使用于整数。

迁移演练

(NOIP 2018 普及组初赛第 9 题) 给定一个含 N 个不相同数字的数组，在最坏情况下，找出其中最大或最小的数，至少需要 $N - 1$ 次比较操作。则最坏情况下，在该数组中同时找最大与最小的数至少需要 () 次比较操作。 ($\lceil \cdot \rceil$ 表示向上取整， $\lfloor \cdot \rfloor$ 表示向下取整)

- A. $\lceil 3N / 2 \rceil - 2$
- B. $\lfloor 3N / 2 \rfloor - 2$
- C. $2N - 2$
- D. $2N - 4$

正确答案: A

解析

在 N 个不相同数字的数组同时查找最大值与最小值的算法思想如下：对 N 个数字两两比较，再将较大的数字与最大值打擂台，较小的数字与最小值打擂台。

最坏情况下的算法时间复杂度：

当 N 是奇数时， $[N / 2] \times 3$ 。将第一个数赋值给最大值和最小值。然后将剩下 $N - 1$ 个整数两两一组，共 $(N - 1) / 2$ 组，每组组内比较一次，与最大值比较一次，与最小值比较一次，共三次。总比较次数： $3 \times (N - 1) / 2$ 。

当 N 是偶数时，将前两个数比较一次，将较大数赋值给最大值、较小数赋值给最小值。然后将剩下的 $N - 2$ 个数两两一组，共 $(N - 2) / 2$ 组，每组组内比较一次，与最大值比较一次，与最小值比较一次，共三次。总比较次数 $3 \times (N - 2) / 2 + 1$ 。

迁移演练

(NOIP 2017 普及组初赛第 17 题) 设 A 和 B 是两个长为 n 的有序数组, 现在需要将 A 和 B 合并成一个排好序的数组, 任何以元素比较作为基本运算的归并算法在最坏情况下至少要做 () 次比较。

- A. n^2
- B. $n \log n$
- C. $2n$
- D. $2n - 1$

正确答案: D

解析: 将 A 与 B 中的元素两两比较, 将值小的放进 C, 直到 C 填满为止。因为 C 有 $2n$ 个空位, 每次两两比较就放进去一个数, 而最后一个剩下的元素可以不用比较直接放进去, 所以一共两两比较了 $2n - 1$ 次。

迁移演练

(NOIP 2015 普及组初赛第 19 题) 某算法的计算时间表示为递推关系式 $T(n)=T(n-1)+n$ (n 为正整数) 及 $T(0)=1$, 则该算法的时间复杂度为()。

- A. $O(\log n)$
- B. $O(n \log n)$
- C. $O(n)$
- D. $O(n^2)$

正确答案: D

解析: $\sum_{i=1}^n i = \frac{n(n+1)}{2}$

第 6 题

设A是n个实数的数组，考虑下面的递归算法：

```
XYZ (A[1..n])  
1.  if n= 1 then return A[1]  
2.  else temp ← XYZ (A[1..n-1])  
3.      if temp < A[n]  
4.      then return temp  
5.      else return A[n]
```

请问算法XYZ的输出是什么？（ ）。

- A. A数组的平均
- B. A数组的最小值
- C. A数组的中值
- D. A数组的最大值

正确答案： B

解析：每次递归均返回前 $n-1$ 个数值的答案与第 n 个数中的较小者，容易发现所求即为 A 数组的最小值。

迁移演练

(CSP 2019 入门组第一轮第 4 题) 若有如下程序段, 其中 s 、 a 、 b 、 c 均已定义为整型变量, 且 a 、 c 均已赋值 (c 大于 0)

```
s = a;
```

```
for (b = 1; b <= c; b++) s = s - 1;
```

则与上述程序段功能等价的赋值语句是 ()

A. $s = a - c;$

B. $s = a - b;$

C. $s = s - c;$

D. $s = b - c;$

正确答案: A

解析: s 的值在 a 的基础上减去了 c 次 1, 即 s 的值变为 $a - c$ 。

迁移演练

(NOIP 2016 普及组初赛第 12 题) 若有如下程序段, 其中s、a、b、c均已定义为整型变量,且a、c均已赋值 (c大于0)

```
s = a;
```

```
for (b = 1; b <= c; b++) s = s + 1;
```

则与上述程序段功能等价的赋值语句是 ()

A. $s = a + b;$

B. $s = a + c;$

C. $s = s + c;$

D. $s = b + c;$

正确答案: **B**

解析: s 的值在 a 的基础上加上了 c 次 1, 即 s 的值变为 $a + c$ 。

迁移演练

(NOIP 2018 普及组初赛第 10 题) 下面的故事与 () 算法有着异曲同工之妙。

从前有座山，山里有座庙，庙里有个老和尚在给小和尚讲故事：
“从前有座山，山里有座庙，庙里有个老和尚在给小和尚讲故事：
‘从前有座山，山里有座庙，庙里有个老和尚给小和尚讲故事……’”

- A. 枚举
- B. 递归
- C. 贪心
- D. 分治

正确答案： B

迁移演练

(NOIP 2018 普及组初赛第 14 题) 为了统计一个非负整数的二进制形式中 1 的个数，代码如下：

```
int CountBit(int x)
{
    int ret = 0;
    while (x)
    {
        ret++;
        _____;
    }
    return ret;
}
```

则空格内要填入的语句是 ()。

- A. $x \gg= 1$
- B. $x \&= x - 1$
- C. $x |= x \gg 1$
- D. $x \ll= 1$

正确答案： B

解析

此题考察对 `lowbit` 函数的理解与应用。

如果一个整数不为0，那么这个整数至少有一位是1。如果我们把这个整数减1，那么原来处在整数最右边的1就会变为0，原来在1后面的所有的0都会变成1(如果最右边的1后面还有0的话)。其余所有位将不会受到影响。

举个例子：一个二进制数1100，从右边数起第三位是处于最右边的一个1。减去1后，第三位变成0，它后面的两位0变成了1，而前面的1保持不变，因此得到的结果是1011。我们发现减1的结果是把最右边的一个1开始的所有位都取反了。这个时候如果我们再把原来的整数和减去1之后的结果做与运算，从原来整数最右边一个1那一位开始所有位都会变成0。如 $1100 \& 1011 = 1000$ 。也就是说，把一个整数减去1，再和原整数做与运算，会把该整数最右边一个1变成0。那么一个整数的二进制有多少个1，就可以进行多少次这样的操作。

迁移演练

(NOIP 2016 普及组初赛第 14 题) 给定含有 n 个不同的数的数组 $L = \langle x_1, x_2, \dots, x_n \rangle$ 。如果 L 中存在 x_i ($1 < i < n$) 使得 $x_1 < x_2 < \dots < x_{i-1} < x_i > x_{i+1} > \dots > x_n$, 则称 L 是单峰的, 并称 x_i 是 L 的“峰顶”。现在已知 L 是单峰的, 请把 a-c 三行代码补全到算法中使得算法 正确找到 L 的峰顶。

a. Search(k+1, n)

b. Search(1, k-1)

c. return L[k]

正确的填空顺序是 ()。

A. c,a,b

B. c,b,a

C. a,b,c

D. b,a,c

正确答案: A

```
Search(1, n)
```

```
1. k ← [n/2]
```

```
2. if L[k] > L[k-1] and L[k] > L[k+1]
```

```
3. then _____
```

```
4. else if L[k] > L[k-1] and L[k] < L[k+1]
```

```
5. then _____
```

```
6. else _____
```


第 7 题

(CSP 2019 入门组第一轮第 6 题、NOIP 2015 普及组初赛第 13 题) 链表不具有的特点是 ()。

- A. 可随机访问任一元素
- B. 不必事先估计存储空间
- C. 插入删除不需要移动元素
- D. 所需空间与线性表长度成正比

正确答案： A

解析

链表是一种物理存储单元上**非连续、非顺序**的存储结构，数据元素的逻辑顺序是通过链表中的指针链接次序实现的。链表由一系列结点（链表中每一个元素称为结点）组成，结点可以在运行时动态生成。每个结点包括两个部分：一个是存储数据元素的数据域，另一个是存储下一个结点地址的指针域。相比于线性表顺序结构，操作复杂。由于不必须按顺序存储，链表在**插入的时候可以达到 $O(1)$ 的复杂度**，比另一种线性表顺序表快得多，但是**查找一个节点或者访问特定编号的节点则需要 $O(n)$ 的时间**，而**线性表和顺序表相应的时间复杂度分别是 $O(\log n)$ 和 $O(1)$** 。使用链表结构可以克服数组链表需要预先知道数据大小的缺点，链表结构可以充分利用计算机内存空间，实现灵活的内存动态管理。但是链表失去了数组随机读取的优点，同时链表由于增加了结点的指针域，空间开销比较大。链表允许插入和移除表上任意位置上的节点，但是不允许随机存取。链表有很多种不同的类型：单向链表，双向链表以及循环链表。

迁移演练

(NOIP 2017 普及组初赛第 13 题) 向一个栈顶指针为 hs 的链式栈中插入一个指针 s 指向的结点时, 应执行 ()。

- A. $hs \rightarrow next = s;$
- B. $s \rightarrow next = hs; hs = s;$
- C. $s \rightarrow next = hs \rightarrow next; hs \rightarrow next = s;$
- D. $s \rightarrow next = hs; hs = hs \rightarrow next;$

正确答案: B

解析: 往栈中插入数据, 栈是向下生长的, 由高地址到低地址, 所以 s 节点就是在低地址处, hs 指向原栈顶, 插入 s 后, s 是新的栈顶, s 的地址比原栈顶 hs 地址低, 所以是 $s \rightarrow next = hs$, 最后 $hs = s$ 更新栈顶指针指向

迁移演练

(NOIP 2015 普及组初赛第 14 题) 线性表若采用链表存储结构, 要求内存中可用存储单元地址()。

- A. 必须连续
- B. 部分地址必须连续
- C. 一定不连续
- D. 连续不连续均可

正确答案: D

第 8 题

有 10 个顶点的无向图至少应该有 () 条边才能确保是一个连通图。

- A. 9
- B. 10
- C. 11
- D. 12

正确答案： A

解析： 有 n 个顶点的无向图至少应该有 $(n - 1)$ 条边才能确保是一个连通图，当恰好有 $(n - 1)$ 条边时所得的连通图称为树。

迁移演练

(NOIP 2018 普及组初赛第 11 题) 由四个没有区别的点构成的简单无向连通图的个数是 ()。

- A. 6
- B. 7
- C. 8
- D. 9

正确答案： A

解析：简单图指没有平行边和自环。



迁移演练

(NOIP 2017 普及组初赛第 10 题) 设 G 是有 n 个结点、 m 条边 ($n \leq m$) 的连通图, 必须删去 G 的 () 条边, 才能使得 G 变成一棵树。

A. $m - n + 1$

B. $m - n$

C. $m + n + 1$

D. $n - m + 1$

正确答案: A

解析: $m - (n - 1) = m - n + 1$

迁移演练

(NOIP 2016 普及组初赛第 15 题) 设简单无向图 G 有 16 条边且每个顶点的度数都是 2, 则图 G 有()个顶点。

- A. 10
- B. 12
- C. 8
- D. 16

正确答案: D

解析: 每个顶点度数都是 2 意味着图 G 恰好成环, 顶点数即边数。

迁移演练

(NOIP 2015 普及组初赛第 12 题) 6 个顶点的连通图的最小生成树, 其边数为()。

- A. 6
- B. 5
- C. 7
- D. 4

正确答案: B

解析: 最小生成树的边数比顶点数少 1。

第 9 题

二进制数 1011 转换成十进制数是 () 。

- A. 11
- B. 10
- C. 13
- D. 12

正确答案： A

解析： $(1011)_2 = 2^0 + 2^1 + 2^3 = 1 + 2 + 8 = (11)_{10}$

迁移演练

(NOIP 2018 普及组初赛第 2 题) 下列四个不同进制的数中, 与其它三项数值上不相等的是

- A. $(269)_{16}$
- B. $(617)_{10}$
- C. $(1151)_8$
- D. $(1001101011)_2$

正确答案: D

解析: $(1001101011)_2 = (619)_{10}$

迁移演练

(NOIP 2017 普及组初赛第 1 题) 在 8 位二进制补码中, 10101011 表示的数是十进制下的 ()。

- A. 43
- B. -85
- C. -43
- D. -84

正确答案: B

解析

- 原码求补码

正整数的补码是其二进制表示，与原码相同

求负整数的补码，将其原码除符号位外的所有位取反（0变1，1变0，符号位为1不变）后加1。

同一个数字在不同的补码表示形式中是不同的。比如-15的补码，在8位二进制中是11110001，然而在16位二进制补码表示中，就是11111111111110001。以下都使用8位2进制来表示。

例：求-5的补码。

-5对应带符号位负数5（10000101）→除符号位外所有位取反（11111010）→加 00000001为（11111011）

所以-5的补码是11111011。

解析

- 补码求原码

已知一个数的补码，求原码的操作其实就是对该补码再求补码：

(1)如果补码的符号位为“0”，表示是一个正数，其原码就是补码。

(2)如果补码的符号位为“1”，表示是一个负数，那么求给定的这个补码的补码就是要求的原码。

例：已知一个补码为11111001，则原码是00000111（7）。

因为符号位为“1”，表示是一个负数，所以该位不变，仍为“1”。

其余七位1111001取反后为0000110；再加1，所以是10000111。

迁移演练

(NOIP 2017 普及组初赛第 15 题) 十进制小数 13.375 对应的二进制数是 ()。

- A. 1101.011
- B. 1011.011
- C. 1101.101
- D. 1010.01

正确答案： A

解析： $(0.375)_{10} = 0.25 + 0.125 = 2^{-2} + 2^{-3} = (0.011)_2$

迁移演练

(NOIP 2016 普及组初赛第 7 题) 二进制数 00101100 和 00010101 的和 () 。

- A. 00101000
- B. 01000001
- C. 01000100
- D. 00111000

正确答案： B

迁移演练

(NOIP 2016 普及组初赛第 8 题) 与二进制小数 0.1 相等的八进制数是 ()。

- A. 0.8
- B. 0.4
- C. 0.2
- D. 0.1

正确答案： B

解析： $(0.1)_2 = (0.5)_{10} = (0.4)_8$

迁移演练

(NOIP 2015 普及组初赛第 4 题) 在计算机内部用来传送、存贮、加工处理的数据或指令都是以()形式进行的。

- A. 二进制码
- B. 八进制码
- C. 十进制码
- D. 智能拼音码

正确答案： A

迁移演练

(NOIP 2015 普及组初赛第 6 题) 二进制数 00100100 和 00010100 的和是()。

- A. 00101000
- B. 01100111
- C. 01000100
- D. 00111000

正确答案： D

迁移演练

(NOIP 2015 普及组初赛第 7 题) 与二进制小数 0.1 相等的十六进制数是()。

- A. 0.8
- B. 0.4
- C. 0.2
- D. 0.1

正确答案： A

解析： $(0.1)_2 = (0.5)_{10} = (0.8)_{16}$

第 10 题

5 个小朋友并排站成一列，其中有两个小朋友是双胞胎，如果要求这两个双胞胎必须相邻，则有（ ）种不同排列方法？

- A. 48
- B. 36
- C. 24
- D. 72

正确答案： A

解析：在做排列的题目时，解决某些元素相邻(要求在一起)问题常用捆绑法：把相邻元素看作一个整体，再与其他元素一起排列，同时注意捆绑元素的内部排列。此题中有 $A_4^4 A_2^2 = 24 \times 2 = 48$ 。

迁移演练

(CSP 2019 入门组第一轮第 7 题) 把 8 个同样的球放在 5 个同样的袋子里，允许有的袋子空着不放，问共有多少种不同的分法？()

提示：如果 8 个球都放在一个袋子里，无论是哪个袋子，都只算同一种分法。

- A. 22
- B. 24
- C. 18
- D. 20

正确答案： C

解析

不妨考虑按袋子数目进行分类讨论。

当放在一个袋子里时： $8 = 8$

当放在两个袋子里时： $8 = 1 + 7, 8 = 2 + 6, 8 = 3 + 5, 8 = 4 + 4$

当放在三个袋子里时： $8 = 1 + 1 + 6, 8 = 1 + 2 + 5, 8 = 1 + 3 + 4,$
 $8 = 2 + 2 + 4, 8 = 2 + 3 + 3$

当放在四个袋子里时： $8 = 1 + 1 + 1 + 5, 8 = 1 + 1 + 2 + 4,$
 $8 = 1 + 1 + 3 + 3, 8 = 1 + 2 + 2 + 3, 8 = 2 + 2 + 2 + 2$

当放在五个袋子里时： $8 = 1 + 1 + 1 + 1 + 4,$
 $8 = 1 + 1 + 1 + 2 + 3, 8 = 1 + 1 + 2 + 2 + 2$

故总的方案数为 $1 + 4 + 5 + 5 + 3 = 18$ 种。

解析

设有 n 个球， m 个盒子。（摘自屈婉玲著《离散数学》）

球区别	盒区别	是否空盒	模型	方案计数
有	有	有	选取	m^n
有	有	无	放球子模型	$m! \left\{ \begin{matrix} n \\ m \end{matrix} \right\}$
有	无	有		$\sum_{k=1}^m \left\{ \begin{matrix} n \\ k \end{matrix} \right\}$
有	无	无		$\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$
无	有	有	不定方程	$C(n+m-1, n)$
无	有	无		$C(n-1, m-1)$
无	无	有	正整数拆分	$G(x) = \frac{1}{(1-x)(1-x^2)\cdots(1-x^m)}, x^n \text{ 系数}$
无	无	无		$G(x) = \frac{x^m}{(1-x)(1-x^2)\cdots(1-x^m)}, x^n \text{ 系数}$

迁移演练

(CSP 2019 入门组第一轮第 12 题) 一副纸牌除掉大小王有 52 张牌，四种花色，每种花色 13 张。假设从这 52 张牌中随机抽取 13 张纸牌，则至少 () 张牌的花色一致。

- A. 4
- B. 2
- C. 3
- D. 5

正确答案： A

解析：根据**抽屉原理**，把 m ($m > n$) 个物体任意放入 n 个抽屉里，若 $m = kn + r$ ，则当 $r > 0$ 时其中一定有某个抽屉至少放 $k + 1$ 个物体。此题中有 $\lceil 13 / 4 \rceil = 4$ 。

第 11 题

(NOIP 2018普及组初赛第 15 题) 下图中所使用的数据结构是 ()。



- A. 栈
- B. 队列
- C. 二叉树
- D. 哈希表

正确答案： A

解析

栈 (stack) 又名堆栈，它是一种运算受限的线性表。限定**仅在表尾进行插入和删除操作**的线性表。这一端被称为栈顶，相对地，把另一端称为栈底。向一个栈插入新元素又称作进栈、入栈或压栈，它是把新元素放到栈顶元素的上面，使之成为新的栈顶元素；从一个栈删除元素又称作出栈或退栈，它是把栈顶元素删除掉，使其相邻的元素成为新的栈顶元素。

队列是一种特殊的线性表，特殊之处在于它只允许在表的前端 (front) **进行删除操作**，而在表的后端 (rear) **进行插入操作**，和栈一样，队列是一种操作受限制的线性表。进行插入操作的端称为队尾，进行删除操作的端称为队头。

迁移演练

(NOIP 2017 普及组初赛第 16 题) 对于入栈顺序为 a, b, c, d, e, f, g 的序列, 下列 () 不可能是合法的出栈序列。

- A. a, b, c, d, e, f, g
- B. a, d, c, b, e, g, f
- C. a, d, b, c, g, f, e
- D. g, f, e, d, c, b, a

正确答案: C

解析: a -> 空 (a 出栈) -> b -> bc -> bcd -> bc (d 出栈), 此时 b 位于栈顶的 c 下方, 不可能先出栈。

迁移演练

(NOIP 2015 普及组初赛第 15 题) 今有一空栈 S，对下列待进栈的数据元素序列 a, b, c, d, e, f 依次进行进栈，进栈，出栈，进栈，进栈，出栈的操作，则此操作完成后，栈 S 的栈顶元素为：

- A. f
- B. c
- C. A
- D. b

正确答案： B

解析： a -> ab -> a -> ac -> acd -> ac

第 12 题

(NOIP 2015 普及组初赛第 17 题) 独根树的高度为 1。具有 61 个结点的完全二叉树的高度为 ()。

- A. 7
- B. 8
- C. 5
- D. 6

正确答案： D

解析：根据美国国家标准技术研究所 (NIST) 的定义，完全二叉树 (Complete Binary Tree)：除最后一层外的每层结点都完全填满，在最后一层上如果不是满的，则只缺少右边的若干结点。

此题中有 $\lfloor \log_2 61 \rfloor = 6$ 。

迁移演练

(CSP 2019 入门组第一轮第 14 题) 假设一棵二叉树的后序遍历序列为DGJHEBIFCA,中序遍历序列为DBGEHJACIF,则其前序遍历序列为 ()。

- A. ABCDEFGHIJ
- B. ABDEGHJCFI
- C. ABDEGJHCFI
- D. ABDEGHJFIC

正确答案： B

解析

整体思路是这样的，由后序遍历找到每个节点，然后由中序遍历判断左右子树，将整个二叉树还原后写出前序遍历。

后序遍历的顺序知道，最后一个A是二叉树的根节点，然后把中序遍历从A分成两段，A左边的是左子树，A右边的是右子树，

结果如下



解析

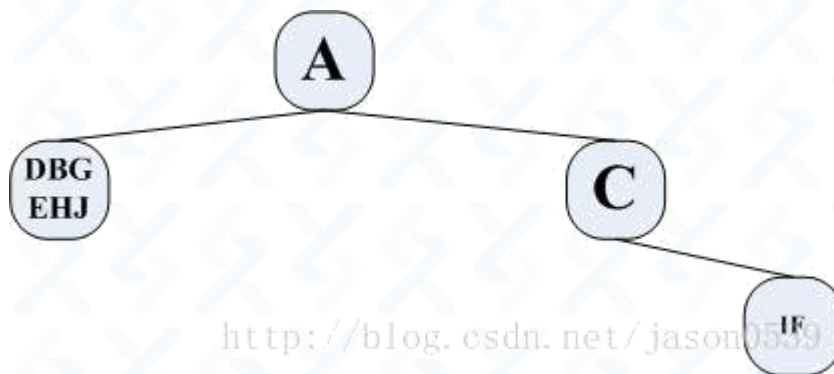
然后看右边的子树，

从后序遍历知道，左子树的后序遍历为IFC,中序遍历为CIF

问题回到刚开始，重复之前的过程，由后序遍历知道根节点为C，
把中序遍历从C分成两段，

左边是左子树，右边是右子树

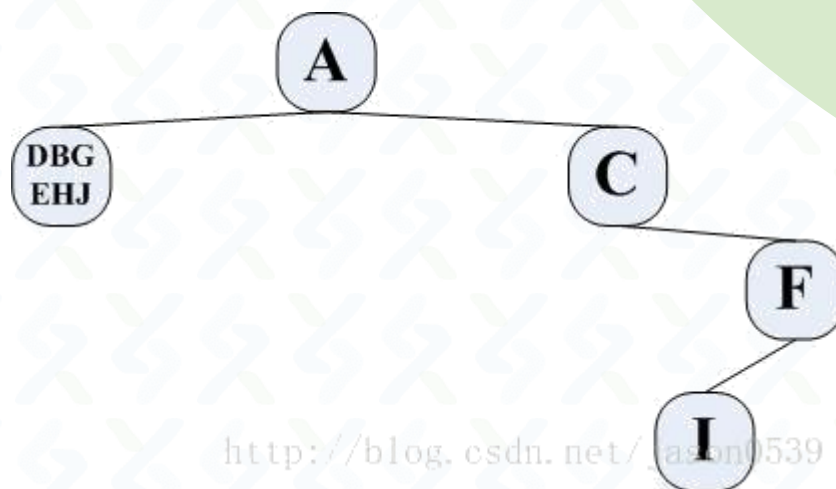
也就是右边只有一个右子树，



解析

然后再次重复以上过程，现在IF的后序遍历是IF,中序遍历是IF，说明

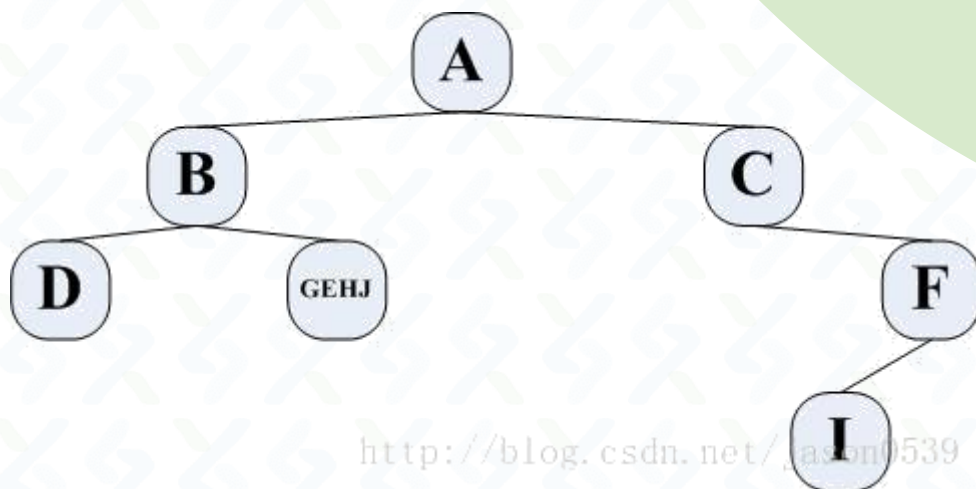
节点时F，I是F的左子树



<http://blog.csdn.net/jason0539>

解析

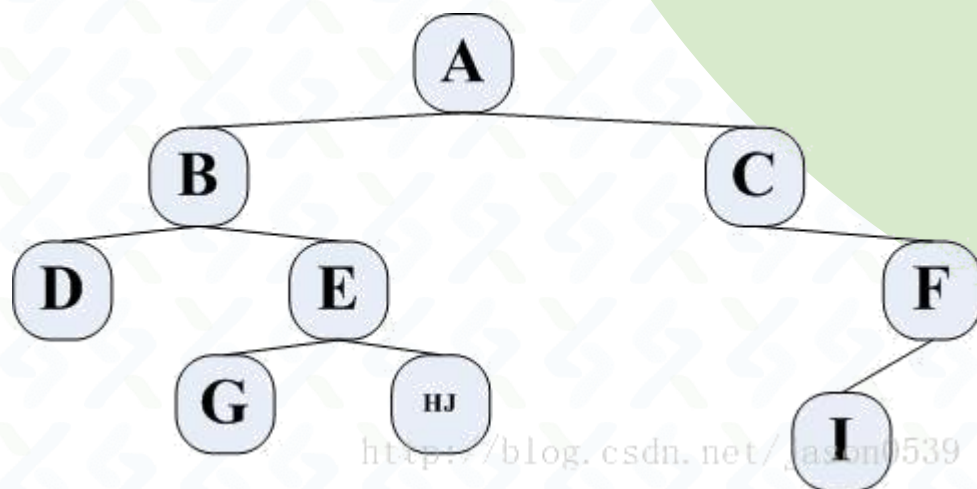
这样，这棵二叉树的右子树就完全复原了，左子树的方法完全相同，就是一个递归过程，流程图如下



<http://blog.csdn.net/jason0539>

解析

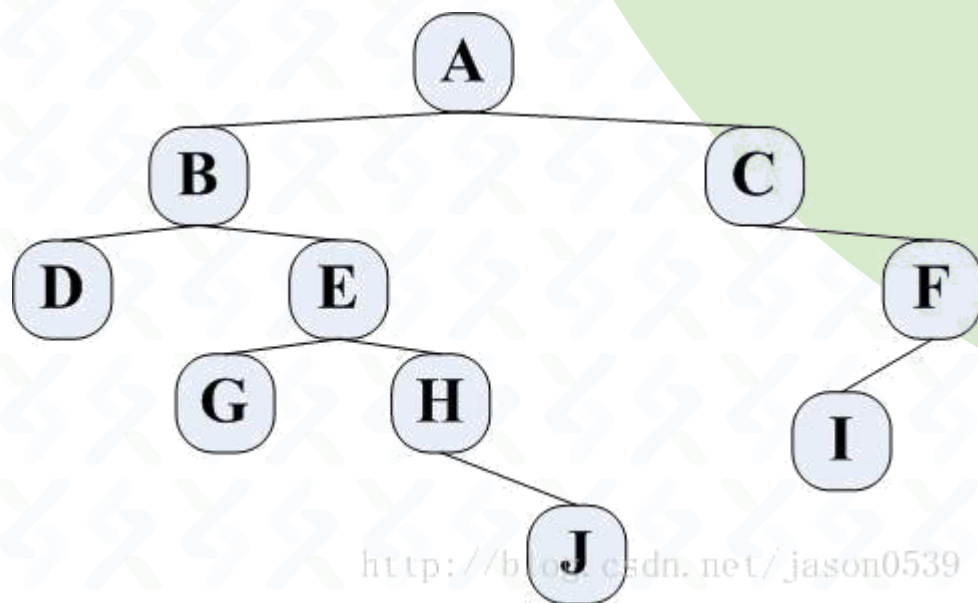
NEXT:



<http://blog.csdn.net/jason0539>

解析

最后得到的完整二叉树如下：



<http://blog.csdn.net/jason0539>

然后写出前序遍历就可以了，是ABDEGHJCFI

迁移演练

(NOIP 2018 普及组初赛第 7 题) 根节点深度为 0, 一棵深度为 h 的满 k ($k > 1$) 叉树, 即除最后一层无任何子节点外, 每一层上的所有结点都有 k 个子结点的树, 共有 () 个结点。

- A. $(k^{h+1}-1)/(k-1)$
- B. k^{h-1}
- C. k^h
- D. $(k^{h-1})/(k-1)$

正确答案: A

解析: $\sum_{i=0}^h k^i = \frac{k^{h+1}-1}{k-1}$

迁移演练

(NOIP 2015 普及组初赛第 16 题) 前序遍历序列与中序遍历序列相同的二叉树为()。

- A. 根结点无左子树
- B. 根结点无右子树
- C. 只有根结点的二叉树或非叶子结点只有左子树的二叉树
- D. 只有根结点的二叉树或非叶子结点只有右子树的二叉树

正确答案： D

解析：“根左右”与“左根右”所得相同， 容易发现左子树应为空。

第 13 题

干支纪年法是中国传统的纪年方法，由10个天干和12个地支组合成60个天干地支。由公历年份可以根据以下公式和表格换算出对应的天干地支。

天干 = (公历年份) 除以10所得余数

地支 = (公历年份) 除以12所得余数

例如，今年是 2020 年，2020 除以 10 余数为 0，查表为“庚”；2020 除以 12，余数为 4，查表为“子”所以今年是庚子年。

请问 1949 年的天干地支是 ()

- A. 己酉
- B. 己亥
- C. 己丑
- D. 己卯

正确答案： C

天干	甲	乙	丙	丁	戊	己	庚	辛	壬	癸		
	4	5	6	7	8	9	0	1	2	3		
地支	子	丑	寅	卯	辰	巳	午	未	申	酉	戌	亥
	4	5	6	7	8	9	10	11	0	1	2	3

解析： $1949 \bmod 10 = 9$ 可得天干为己； $1949 \bmod 12 = 5$ 可得地支为丑。

迁移演练

(NOIP 2018 普及组初赛第 6 题) 如果开始时计算机处于小写输入状态, 现在有一只小老鼠反复按照CapsLock、字母键A、字母键 S、字母键D、字母键 F 的顺序循环按键, 即 CapsLock、A、S、D、F、CapsLock、A、S、D、F、……, 屏幕上输出的第 81 个字符是字母 ()

- A. A
- B. S
- C. D
- D. a

正确答案: A

解析: $81 \bmod 4 = 1$, 输出的第 81 个字符与第 1 个字符相同。

迁移演练

(NOIP 2017 普及组初赛第 8 题) 2017 年 10 月 1 日是星期日, 1999 年 10 月 1 日是 ()。

- A. 星期三
- B. 星期日
- C. 星期五
- D. 星期二

正确答案: C

解析: 2000是闰年, 2004, 2008, 2012, 2016年应该者是闰年, 共5个, 即5个366天, 13个非闰年, 共
 $13 \times 365 + 5 \times 366 = 4745 + 1830 = 6575$ 天, $6575 \% 7 = 939$ 余2, 因1999年是向前找, 所以是星期五

第 14 题

10 个三好学生名额分配到 7 个班级，每个班级至少有一个名额，一共有（ ）种不同的分配方案。

- A. 84
- B. 72
- C. 56
- D. 504

正确答案： A

解析：插板法，10 个名额可以插入 9 个，7 个班级插入 6 个板子，即 $C_9^6 = 84$ 。

迁移演练

(NOIP 2018 普及组初赛第 12 题) 设含有 10 个元素的集合的全部子集数为 S ，其中由 7 个元素组成的子集数为 T ，则 T / S 的值为 ()。

- A. $5 / 32$
- B. $15 / 128$
- C. $1 / 8$
- D. $21 / 128$

正确答案： B

解析： $S = 2^{10} = 1024$ ， $T = C_{10}^7 = 120$

迁移演练

(NOIP 2017 普及组初赛第 9 题) 甲、乙、丙三位同学选修课程，从 4 门课程中，甲选修 2 门，乙、丙各选修 3 门，则不同的选修方案共有 () 种。

- A. 36
- B. 48
- C. 96
- D. 192

正确答案： C

解析： $C_4^2 \times C_4^3 \times C_4^3 = 6 \times 4 \times 4 = 96$

第 15 题

有五副不同颜色的手套（共 10 只手套，每副手套左右手各 1 只），一次性从中取 6 只手套，请问恰好能配成两副手套的不同取法有（ ）种。

- A. 120
- B. 180
- C. 150
- D. 30

正确答案： A

解析

从 5 副不同颜色的手套取 6 只手套，恰好能配成 2 副手套，可以分 3 步解决：

从 5 副手套中挑出 2 副手套，方案数有： $C_5^2 = 10$

从剩下的 3 副手套中挑出 2 副手套（剩下的 2 只手套不同，所以需要 2 副手套中选出），方案数有： $C_3^2 = 3$

由于颜色不同，这 2 只手套一共有 4 种组合：左左、左右、右左、右右。利用乘法原理，一共有 120 种方案。

迁移演练

(NOIP 2017 普及组初赛第 14 题) 若串 $S = \text{"copyright"}$, 其子串的个数是 ()。

- A. 72
- B. 45
- C. 46
- D. 36

正确答案: C

解析: $1 + \sum_{i=1}^n i = \frac{n(n+1)}{2} + 1$

迁移演练

(NOIP 2016 普及组初赛第 16 题) 有7个一模一样的苹果，放到3个一样的盘子中，一共有 () 种放法。

- A. 7
- B. 8
- C. 21
- D. 3^7

正确答案： B

解析： 各种可能的放置情况如下： $(7, 0, 0)$, $(6, 1, 0)$,
 $(5, 2, 0)$, $(5, 1, 1)$, $(4, 3, 0)$, $(4, 2, 1)$,
 $(3, 3, 1)$, $(3, 2, 2)$, 共8种

二、阅读程序

程序输入不超过数组或字符串定义的范围；判断题正确填 $\sqrt{\quad}$ ，错误填 \times 。除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分

第 16 题

```
#include <cstdlib>
#include <iostream>
using namespace std;

char encoder[26] = {'C', 'S', 'P', 0};
char decoder[26];

string st;

int main() {
    int k = 0;
    for (int i = 0; i < 26; ++i)
        if (encoder[i] != 0) ++k;
    for (char x = 'A'; x <= 'Z'; ++x) {
        bool flag = true;
        for (int i = 0; i < 26; ++i)
            if (encoder[i] == x) {
                flag = false;
                break;
            }
        if (flag) {
            encoder[k] = x;
            ++k;
        }
    }
    for (int i = 0; i < 26; ++i)
        decoder[encoder[i] - 'A'] = i + 'A';
    cin >> st;
    for (int i = 0; i < st.length(); ++i)
        st[i] = decoder[st[i] - 'A'];
    cout << st;
    return 0;
}
```

• 判断题

- 1) 输入的字符串应当只由大写字母组成，否则在访问数组时可能越界。 ()
- 2) 若输入的字符串不是空串，则输入的字符串与输出的字符串一定不一样。 ()
- 3) 将第 12 行的“ $i < 26$ ”改为“ $i < 16$ ”，程序运行结果不会改变。 ()
- 4) 将第 26 行的“ $i < 26$ ”改为“ $i < 16$ ”，程序运行结果不会改变。 ()

正确答案： √ × √ ×

解析

本题主要实现字符串加密，代码中的两个字符数组 encoder 与 decoder 分别对应编码与解码。主要功能是用字符 'C'、'S'、'P' 替换字母表中的 'A'、'B'、'C'，其余的字符向后移动。如表所示，encoder 列即为加密状态；根据 $\text{decoder}[\text{encoder}[i] - 'A'] = i + 'A'$ 的规则解密为 decoder 状态。当输入的字符串 st 经过加密，从 decoder 相应得到解密后的 st 字符串。反之从 decoder 的字母，也能比照出 cin 的字母。

i	cin	decoder	encoder
0	A	D	C
1	B	E	S
2	C	A	P
3	D	F	A
4	E	G	B
5	F	H	D
6	G	I	E
7	H	J	F
8	I	K	G
9	J	L	H
10	K	M	I
11	L	N	J
12	M	O	K
13	N	P	L
14	O	Q	M
15	P	C	N
16	Q	R	O
17	R	S	Q
18	S	B	R
19	T	T	T
20	U	U	U
21	V	V	V
22	W	W	W
23	X	X	X
24	Y	Y	Y
25	Z	Z	Z

解析

判断 1：如果字符对应的 ASCII 码比大写字母 A 小（即减去 'A'），则下标可能会出现负值，最终导致越界，所以下标应限制在大写字母范围内，故本描述正确。

判断 2：参照表，加密与解密序列中大写字母 T 之后的字母完全一致，即输入“UVWXYZ”输出也是“UVWXYZ”，故本描述有误。

判断 3：初始状态加密只对应 3 个有效字母，条件只需满足 i 不大于 3 即可，最终结果不受影响，故本描述正确。

判断 4：遍历（循环）范围需包含所有的大写字母，故本描述有误。

第 16 题

```
#include <cstdlib>
#include <iostream>
using namespace std;

char encoder[26] = {'C', 'S', 'P', 0};
char decoder[26];

string st;

int main() {
    int k = 0;
    for (int i = 0; i < 26; ++i)
        if (encoder[i] != 0) ++k;
    for (char x = 'A'; x <= 'Z'; ++x) {
        bool flag = true;
        for (int i = 0; i < 26; ++i)
            if (encoder[i] == x) {
                flag = false;
                break;
            }
        if (flag) {
            encoder[k] = x;
            ++k;
        }
    }
    for (int i = 0; i < 26; ++i)
        decoder[encoder[i] - 'A'] = i + 'A';
    cin >> st;
    for (int i = 0; i < st.length(); ++i)
        st[i] = decoder[st[i] - 'A'];
    cout << st;
    return 0;
}
```

• 单选题

5) 若输出的字符串为“ABCABCABCA”，则下列说法正确的是（ ）。

- A. 输入的字符串中既有S又有P
- B. 输入的字符串中既有S又有B
- C. 输入的字符串中既有A又有P
- D. 输入的字符串中既有A又有B

正确答案：A

解析：依据表，输出“ABC”对应输入“CSP”，满足既有 S 又有 P。

第 16 题

```
#include <cstdlib>
#include <iostream>
using namespace std;

char encoder[26] = {'C', 'S', 'P', 0};
char decoder[26];

string st;

int main() {
    int k = 0;
    for (int i = 0; i < 26; ++i)
        if (encoder[i] != 0) ++k;
    for (char x = 'A'; x <= 'Z'; ++x) {
        bool flag = true;
        for (int i = 0; i < 26; ++i)
            if (encoder[i] == x) {
                flag = false;
                break;
            }
        if (flag) {
            encoder[k] = x;
            ++k;
        }
    }
    for (int i = 0; i < 26; ++i)
        decoder[encoder[i] - 'A'] = i + 'A';
    cin >> st;
    for (int i = 0; i < st.length(); ++i)
        st[i] = decoder[st[i] - 'A'];
    cout << st;
    return 0;
}
```

• 单选题

6) 若输出的字符串为“CSPCSPCSPCSP”，则下列说法正确的是（ ）。

- A. 输入的字符串中既有P又有K
- B. 输入的字符串中既有J又有R
- C. 输入的字符串中既有J又有K
- D. 输入的字符串中既有P又有R

正确答案：A

解析：同上表，输出“CSP”对应输入“PRN”，满足既有 P 又有 R。

第 17 题

```
#include <iostream>
using namespace std;

long long n, ans;
int k, len;
long long d[1000000];

int main() {
    cin >> n >> k;
    d[0] = 0;
    len = 1;
    ans = 0;
    for (long long i = 0; i < n; ++i) {
        ++d[0];
        for (int j = 0; j + 1 < len; ++j) {
            if (d[j] == k) {
                d[j] = 0;
                d[j + 1] += 1;
                ++ans;
            }
        }
        if (d[len - 1] == k) {
            d[len - 1] = 0;
            d[len] = 1;
            ++len;
            ++ans;
        }
    }
    cout << ans << endl;
    return 0;
}
```

假设输入的 n 是不超过 2^{62} 的正整数， k 都是不超过 10000 的正整数，完成下面的判断题和单选题：

- 判断题

- 1) 若 $k=1$ ，则输出 ans 时， $len=n$ 。 ()
- 2) 若 $k>1$ ，则输出 ans 时， len 一定小于 n 。 ()
- 3) 若 $k>1$ ，则输出 ans 时， k^{len} 一定大于 n 。 ()

正确答案： $\times \times \checkmark$

解析

本题是进制转换问题，模拟 k 进制的第 n 个数。即询问将某数字进行进制转换后的进位计数。代码中 len 表示当前进位数， ans 计数器记录发生了几次进位。

判断 1：应该为第 2 位，即 len 为 2，故本描述有误。

判断 2：题意为判断 k 进制的位数是否一定小于 n ，以 2 为例，2 的二进制是 10，它的位数是 2， $len(2 \text{ 位})$ 并不小于 n ，故本题不正确。

判断 3：一个 k 进制数，如果有 len 位，每一位有 k 种变化，一共能表示 k^{len} 种数值，数值是从 0 到 $k^{len} - 1$ 的，故本描述正确。

第 17 题

```
#include <iostream>
using namespace std;

long long n, ans;
int k, len;
long long d[1000000];

int main() {
    cin >> n >> k;
    d[0] = 0;
    len = 1;
    ans = 0;
    for (long long i = 0; i < n; ++i) {
        ++d[0];
        for (int j = 0; j + 1 < len; ++j) {
            if (d[j] == k) {
                d[j] = 0;
                d[j + 1] += 1;
                ++ans;
            }
        }
        if (d[len - 1] == k) {
            d[len - 1] = 0;
            d[len] = 1;
            ++len;
            ++ans;
        }
    }
    cout << ans << endl;
    return 0;
}
```

• 单选题

4) 若输入的 n 等于: 10^{15} , 输入的 k 为 1, 则输出等于 ()。

- A. 1
- B. $(10^{30} - 10^{15})/2$
- C. $(10^{30} + 10^{15})/2$
- D. 10^{15}

正确答案: D

解析: 输入数为 1 的情况, 每个 i 会发生一次进位, 共循环 n 次, 因此共产生 n 次进位。

第 17 题

```
#include <iostream>
using namespace std;

long long n, ans;
int k, len;
long long d[1000000];

int main() {
    cin >> n >> k;
    d[0] = 0;
    len = 1;
    ans = 0;
    for (long long i = 0; i < n; ++i) {
        ++d[0];
        for (int j = 0; j + 1 < len; ++j) {
            if (d[j] == k) {
                d[j] = 0;
                d[j + 1] += 1;
                ++ans;
            }
        }
        if (d[len - 1] == k) {
            d[len - 1] = 0;
            d[len] = 1;
            ++len;
            ++ans;
        }
    }
    cout << ans << endl;
    return 0;
}
```

• 单选题

5) 若输入的 n 等于 205,891,132,094,649 (即 3^{30})，输入的 k 为 3，则输出等于 ()。

- A. 3^{30}
- B. $(3^{30}-1)/2$
- C. $3^{30}-1$
- D. $(3^{30}+1)/2$

正确答案: B

解析

题目要计算的是将 n 转换为3进制的过程中，进位次数 ans ，其中 n 为 3^{30} 。由于数据规模较大，不妨先计算 3^3 ，寻找规律。

$3^3=(1000)_3$ ，为求进位次数，可以将进位情况分为以下几类：

- 向最高位进位，有1种情况： $(222)_3$
- 向次高位进位，有3种情况： $(022)_3$ 、 $(122)_3$ 、 $(222)_3$
- 向第二位进位，有9种情况： $(002)_3$ 、 $(012)_3$ 、 $(022)_3$ 、 $(102)_3$ 、 $(112)_3$ 、 $(122)_3$ 、 $(202)_3$ 、 $(212)_3$ 、 $(222)_3$

因此，转换过程中总的进位次数 $ans = 1 + 3 + 9 = 13$ 。

进一步分析发现，对 3^m 的进位情况进行分类，每类的进位次数满足公比为3的等比数列，因此总的进位次数 $ans = 3^0 + 3^1 + 3^2$

$$+ \dots + 3^{m-1} = 1 \times \frac{1-3^m}{1-3} = \frac{3^m-1}{2}。当m=30时，ans = \frac{3^{30}-1}{2}。$$

进步一推广，当输入 $n=k^m$ 时， $ans = \frac{k^m-1}{k-1}$

第 17 题

```
#include <iostream>
using namespace std;

long long n, ans;
int k, len;
long long d[1000000];

int main() {
    cin >> n >> k;
    d[0] = 0;
    len = 1;
    ans = 0;
    for (long long i = 0; i < n; ++i) {
        ++d[0];
        for (int j = 0; j + 1 < len; ++j) {
            if (d[j] == k) {
                d[j] = 0;
                d[j + 1] += 1;
                ++ans;
            }
        }
        if (d[len - 1] == k) {
            d[len - 1] = 0;
            d[len] = 1;
            ++len;
            ++ans;
        }
    }
    cout << ans << endl;
    return 0;
}
```

• 单选题

6) 若输入的 n 等于 100,010,002,000,090, 输入的 k 为 10, 则输出等于 ()。

- A. 11,112,222,444,543
- B. 11,122,222,444,453
- C. 11,122,222,444,543
- D. 11,112,222,444,453

正确答案: D

解析

利用26题得到的结论，那么当 $k = 10$ 时：

- 累加到 100,000,000,000,000, 需要进位 11,111,111,111,111 次
- 累加到 10,000,000,000, 需要进位 1,111,111,111,111 次
- 累加到 2,000,000, 需要进位 $2 \times 111,111 = 222,222$ 次
- 累加到 90, 需要进位 $9 \times 1 = 9$ 次

总的进位次数 $\text{ans} = 11,111,111,111,111 + 1,111,111,111,111 + 222,222 + 9 = 11,112,222,444,453$

第 18 题

```
#include <algorithm>
#include <iostream>
using namespace std;

int n;
int d[50][2];
int ans;

void dfs(int n, int sum) {
    if (n == 1) {
        ans = max(sum, ans);
        return;
    }
    for (int i = 1; i < n; ++i) {
        int a = d[i - 1][0], b = d[i - 1][1];
        int x = d[i][0], y = d[i][1];
        d[i - 1][0] = a + x;
        d[i - 1][1] = b + y;
        for (int j = i; j < n - 1; ++j)
            d[j][0] = d[j + 1][0], d[j][1] = d[j + 1][1];
```

```
        int s = a + x + abs(b - y);
        dfs(n - 1, sum + s);
        for (int j = n - 1; j > i; --j)
            d[j][0] = d[j - 1][0], d[j][1] = d[j - 1][1];
        d[i - 1][0] = a, d[i - 1][1] = b;
        d[i][0] = x, d[i][1] = y;
    }
}

int main() {
    cin >> n;
    for (int i = 0; i < n; ++i)
        cin >> d[i][0];
    for (int i = 0; i < n; ++i)
        cin >> d[i][1];
    ans = 0;
    dfs(n, 0);
    cout << ans << endl;
    return 0;
}
```


解析

本题类似于“石子合并”，针对一个 2 列的二维数组，不断选出相邻的两行进行合并，直到合并为 1 行为止。结果为每次合并时相邻两行第 0 列之和与第 1 列之差的绝对值，最后合并求最值。其中，ans 为和最大的方案，本题主要考察考生的搜索功底。

第 18 题

假设输入的 n 是不超过50的正整数， $d[i][0]$ 、 $d[i][i]$ 都是不超过10000的正整数，完成下面的判断题和单选题：

- 判断题

- 1) 若输入 n 为 0,此程序可能会死循环或发生运行错误。 ()
- 2) 若输入 n 为 20,接下来的输入全为 0, 则输出为 0。 ()
- 3) 输出的数一定不小于输入的 $d[i][0]$ 和 $d[i][i]$ 的任意一个。 ()

正确答案： × √ ×

解析

判断 1：当 n 为0时，dfs中for循环不会被执行，因此输出0，不会发生死循环。

判断 2：当两个序列同时为0，dfs中sum也始终为0，则输出为0。

判断 3：举出一个反例即可，例如 $n = 1$ 时， $ans = 0$ ，小于 $d[1][0]$ 和 $d[1][1]$ 。

第 18 题

• 单选题

4) 若输入的 n 为 20,接下来输入 20 个 9 和 20 个 0,则输出为 ()。

A. 1890

B. 1881

C. 1908

D. 1917

正确答案： B

解析：若B序列为0，那么dfs就是对A序列进行迭代求和。近似贪心思想，每次让刚合并过的行继续合并会得到更大的 sum，即 $(9 + 9) + (9 + 9 + 9) + (9 + 9 + 9 + 9) + \dots$ ，因此 $\text{ans} = \frac{(2+20) \times 19}{2} \times 9$

第 18 题

- 单选题

5) 若输入的 n 为 30,接下来输入 30 个 0 和 30 个 5,则输出为 ()。

- A. 2000
- B. 2010
- C. 2030
- D. 2020

正确答案： C

解析：若A序列为0，那么dfs就是对B序列相邻项差的绝对值迭代求和。带入 n 较小的情况可以发现规律，如 $n = 4$ 时加到 $2 * 5$ ，可以推出 $n = 30$ 时加到 $28 * 5$ ，即 $0 + 5 + 10 + \dots + 28 * 5 = (28 * 29) / 2 = 2030$

第 18 题

- 单选题

6) (4分) 若输入的 n 为 15，接下来的输入是 15 到 1，以及 15 到 1，则输出为 ()。

- A. 2440
- B. 2220
- C. 2240
- D. 2420

正确答案： C

解析

计算量很大，从左往右贪心，将 n 取不同的值分析规律

n	15	14	13	12	...	2
b[i] [0]	15+14=29	15+14+13=42	15+14+13+12=54	15+14+13+12+11=65	...	15+14+13+12+11+2+1
s	15+14+15-14= 30	(15+14)+13+(15+14)-13=6	(15+14+13)+12+(15+14+13)-12=84	(15+14+13+12)+11+(15+14+13+12)-11=130	...	(15+14+...+2)-1+(15+14+...+2)-1=238
ans	0+30=30	30+58=88	88+84=172	172+108=280	...	15×2+(15+14)×2+...+(15+14+...+2)×2

因此，
$$\begin{aligned} \text{ans} &= 15 \times 2 + (15 + 14) \times 2 + \dots + (15 + 14 + \dots + 2) \times 2 \\ &= (15 + 29 + 42 + 54 + 65 + 75 + 84 + 92 + 99 + 105 + 110 + 114 + 117 + 119) \times 2 = 1120 \times 2 = 2240 \end{aligned}$$

三、完善程序

单选题，每小题 3 分，共计 30 分

第 19 题

1. (质因数分解) 给出正整数 n ，请输出将 n 质因数分解的结果，结果从小到大输出。例如：输入 $n=120$ ，程序应该输出 2 2 2 3 5，表示： $120 = 2 \times 2 \times 2 \times 3 \times 5$ 。输入保证 $2 \leq n \leq 10^9$ 。提示：先从小到大枚举变量 i ，然后用 i 不停试除 n 来寻找所有的质因子。

试补全程序。

解析： 本题依靠枚举，输出质因数分解的结果。

第 19 题

```
#include <stdio>
using namespace std;
int n, i;

int main() {
    scanf("d", &n);
    for(i = ①; ② <=n; i ++){
        ③{
            printf("%d ", i);
            n = n / i;
        }
    }
    if(④)
        printf("%d ", ⑤);
    return 0;
}
```

1) ①处应填 ()

- A. 1
- B. n-1
- C. 2
- D. 0

正确答案: C

解析: 枚举质因子最小值以 2 为起点。

第 19 题

```
#include <stdio>
using namespace std;
int n, i;

int main() {
    scanf("d", &n);
    for(i = ①; ② <=n; i ++){
        ③{
            printf("%d ", i);
            n = n / i;
        }
    }
    if(④)
        printf("%d ", ⑤);
    return 0;
}
```

2) ②处应填 ()

- A. n/i
- B. $n/(i*i)$
- C. $i*i$
- D. $i*i*i$

正确答案: C

解析: 质因子范围访问至 \sqrt{n} 即可, 无需完整遍历。

第 19 题

```
#include <stdio>
using namespace std;
int n, i;

int main() {
    scanf("d", &n);
    for(i = ①; ② <=n; i ++){
        ③{
            printf("%d ", i);
            n = n / i;
        }
    }
    if(④)
        printf("%d ", ⑤);
    return 0;
}
```

3) ③处应填 ()

- A. if(n%i==0)
- B. if(i*i<=n)
- C. while(n%i==0)
- D. while(i*i<=n)

正确答案: C

解析: 对因子相同的情况, 不断迭代直到不能被 i 整除为止。

第 19 题

```
#include <stdio>
using namespace std;
int n, i;

int main() {
    scanf("d", &n);
    for(i = ①; ② <=n; i ++){
        ③{
            printf("%d ", i);
            n = n / i;
        }
    }
    if(④)
        printf("%d ", ⑤);
    return 0;
}
```

4) ④处应填 ()

- A. $n > 1$
- B. $n \leq 1$
- C. $i < n/i$
- D. $i + i \leq n$

正确答案: A

解析: 如果 $n > 1$, 就表示因子未被完全除尽, 即说明 n 中包含一个不小于 \sqrt{n} 的质因子。

第 19 题

```
#include <stdio>
using namespace std;
int n, i;

int main() {
    scanf("d", &n);
    for(i = ①; ② <=n; i ++){
        ③{
            printf("%d ", i);
            n = n / i;
        }
    }
    if(④)
        printf("%d ", ⑤);
    return 0;
}
```

5) ⑤处应填 ()

- A. 2
- B. n/i
- C. n
- D. i

正确答案: C

解析: 因为其它小于等于 \sqrt{n} 的质因子已从 n 中除掉, 最后还剩下 n 未输出, 所以输出 n 。

第 20 题

2. (最小区间覆盖) 给出 n 个区间, 第 i 个区间的左右端点是 $[a_i, b_i]$ 。现在要在这些区间中选出若干个, 使得区间 $[0, m]$ 被所选区间的并覆盖 (即每一个 $0 \leq i \leq m$ 都在某个所选的区间中)。保证答案存在, 求所选区间个数的最小值。

输入第一行包含两个整数 n 和 m ($1 \leq n \leq 5000, 1 \leq m \leq 10^9$) 接下来 n 行, 每行两个整数 a_i, b_i ($0 \leq a_i, b_i \leq m$)。

提示: 使用贪心法解决这个问题。先用 $O(n^2)$ 的时间复杂度排序, 然后贪心选择这些区间。

试补全程序。

第 20 题

```
#include <iostream>

using namespace std;

const int MAXN = 5000;
int n, m;
struct segment { int a, b; } A[MAXN];

void sort() // 排序
{
    for (int i = 0; i < n; i++)
        for (int j = 1; j < n; j++)
            if (①)
            {
                segment t = A[j];
                ②
            }
}
```

```
int main()
{
    cin >> n >> m;
    for (int i = 0; i < n; i++)
        cin >> A[i].a >> A[i].b;
    sort();
    int p = 1;
    for (int i = 1; i < n; i++)
        if (③)
            A[p++] = A[i];
    n = p;
    int ans = 0, r = 0;
    int q = 0;
    while (r < m)
    {
        while (④)
            q++;
        ⑤;
        ans++;
    }
    cout << ans << endl;
    return 0;
}
```


解析

算法思想如下：

- 首先使用了冒泡排序的思想，将所有区间按照左端点从小到大排序。
- 去除所有被包含的区间，例如 $[3,10]$ 、 $[4,9]$ ，后面的区间被前面的区间包含了，不需要进行判断。
- 为覆盖区间 $[0,m]$ ，将 r 设为 0，在区间数组中找到离 r 最近的、并且左端点 $\leq r$ 的区间 q ，使用区间 q 的右端点更新 r ，重复这个过程直到 $r \geq m$ 。

第 20 题

1)①处应填 ()

- A. $A[j].b > A[j-1].b$
- B. $A[j].a < A[j-1].a$
- C. $A[j].a > A[j-1].a$
- D. $A[j].b < A[j-1].b$

正确答案： B

解析：对区间内的第一个元素进行冒泡排序，依据题意进行升序排序，如果右边的起点小于左边的起点则交换。

第 20 题

2)②处应填 ()

- A. $A[j+1]=A[j];A[j]=t;$
- B. $A[j-1]=A[j];A[j]=t;$
- C. $A[j]=A[j+1];A[j+1]=t;$
- D. $A[j]=A[j-1];A[j-1]=t;$

正确答案： D

解析：交换顺序，通过第三方变量中继传递，即将 $A[j]$ 传递给 t ，然后将 $A[j-1]$ 赋值给 $A[j]$ ，最后 t 的值再传递给 $A[j-1]$ 。

第 20 题

3)③处应填 ()

- A. $A[i].b > A[p-1].b$
- B. $A[i].b < A[i-1].b$
- C. $A[i].b > A[i-1].b$
- D. $A[i].b < A[p-1].b$

正确答案：A

解析：去掉被包含的区间，只有当第*i*个和第*p*-1个区间不是包含关系时，将第*i*个区间放入第*p*个位置

第 20 题

4)④处应填 ()

- A. $q+1 < n \&\& A[q+1].a \leq r$
- B. $q+1 < n \&\& A[q+1].b \leq r$
- C. $q < n \&\& A[q].a \leq r$
- D. $q < n \&\& A[q].b \leq r$

正确答案：A

解析：变量 q 从 0 开始，当 $q + 1$ 小于 n 时，才能确保第 $q + 1$ 个区间有值，同时只有在 $q + 1$ 个区间的第一个数小等于 r 时，这个区间才起用。

第 20 题

5)⑤处应填 ()

- A. $r = \max(r, A[q+1].b)$
- B. $r = \max(r, A[q].b)$
- C. $r = \max(r, A[q+1].a)$
- D. $q++$

正确答案： B

解析：当前存储的最右边的区间和第 q 个区间的第二个元素取最大值，即用区间 q 的右端点更新 r 。