



# CSP-S1 2022 讲评

一扶苏一  
山东大学  
洛谷网络科技有限公司

# 选择题

# 选择题答案

---

BADCA

BCBDA

CDBBB

# 第一题

在 Linux 系统终端中，用于切换工作目录的命令为：

A. ls

B. cd

C. cp

D. all

ls：列出当前目录下的内容

cp：复制某个文件

## 第二题

你同时用 `time` 命令和秒表为某个程序在单核 CPU 的运行计时。假如 `time` 命令的输入是：

```
real  0m30.721s
```

```
user  0m24.579s
```

```
sys   0m6.123s
```

以下最接近秒表时的时长是：

A. 30s

B. 24s

C. 18s

D. 6s

## 第二题

---

CPU 执行程序时共有两个状态：用户态和内核态。用于执行不同类型的指令。

user 是用户态用时，sys 是内核态用时。real 是程序实际用时。

## 第三题

若元素 a、b、c、d、e、f 依次进栈，允许进栈、退栈操作交替进行，但不允许连续三次退栈操作，则不可能得到的出栈序列是（ ）。

- A. dcebfa
- B. cbdaef
- C. bcaefd
- D. afedcb

栈里的元素肯定是从小到大的。连续三次退栈会产生一个长度为 3 的连续子段。找存在这样连续子段的选项即可。



## 第四题

考虑对  $n$  个数进行排序，以下最坏时间复杂度低于  $O(n^2)$  的排序方法是（ ）。

- A. 插入排序
- B. 冒泡排序
- C. 归并排序
- D. 快速排序

只有 C 是稳定的  $O(n \log n)$ ，其余的最坏情况均为  $\Theta(n^2)$ 。



## 第五题

假设在基数排序过程中，受宇宙射线的影响，某项数据异变为一个完全不同的值。请问排序算法结束后，可能出现的最坏情况是（ ）。

- A. 移除受影响的数据后，最终序列是有序序列
- B. 移除受影响的数据后，最终序列是前后两个有序的子序列
- C. 移除受影响的数据后，最终序列是一个有序的子序列和一个基本无序的子序列
- D. 移除受影响的数据后，最终序列基本无序

可以类比计数排序。每个数据只影响它自己。

## 第六题

计算机系统用小端（Little Endian）和大端（Big Endian）来描述多字节数据的存储地址顺序模式，其中小端表示将低位字节数据存储在低地址的模式、大端表示将高位字节数据存储在低地址的模式。在小端模式的系统和大端模式的系统分别编译和运行以下C++代码段表示的程序，将分别输出什么结果？（ ）

```
unsigned x = 0xDEADBEEF;  
unsigned char *p = (unsigned char *)&x;  
printf("%X", *p);
```

EF、DE

## 第六题

内存可以看作一个长数组，每位 1 Byte（8 bit）。地址就是数组下标。

一个 unsigned（4 Byte）总是存在连续四个内存块里。

指针总是指向低地址（较小的下标）

小端：

EF	BE	AD	DE
----	----	----	----

&x

大端：

DE	AD	BE	EF
----	----	----	----

把 &x 强转成 char\* 其实是只取一个 Byte 的值。所以分别是 EF 和 DE。

## 第七题

一个深度为 5（根结点深度为 1）的完全 3 叉树，按前序遍历的顺序给结点从 1 开始编号，则第 100 号结点的父结点是第（ ）号。

- A. 95
- B. 96
- C. 97
- D. 98

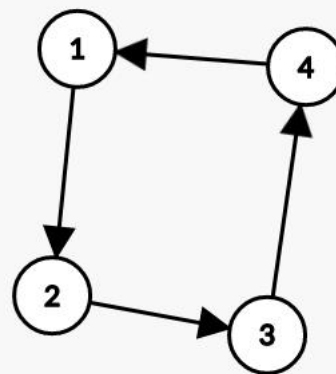
根节点的左子树共有  $3^0 + 3^1 + 3^2 + 3^3 = 40$  个点。同理中子树也有 40 个点。所以右子树根节点的编号是  $1+40+40+1=82$ 。然后手动画一下接下来 18 个结点即可。也可以用类似的方式再算一层。

## 第八题

强连通图的性质不包括（ ）：

- A. 每个顶点的度数至少为 1
- B. 任意两个顶点之间都有边相连
- C. 任意两个顶点之间都有路径相连
- D. 每个顶点都至少连有一条边

强连通图的定义只要求了任意两点间均有路径可达。

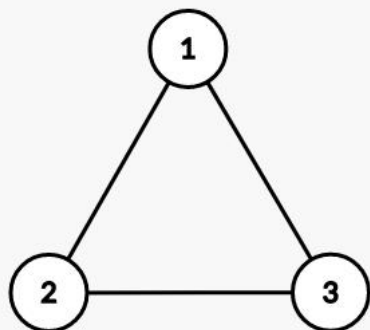


## 第九题

每个顶点度数均为 2 的无向图称为“2 正规图”。由编号为从 1 到  $n$  的顶点构成的所有 2 正规图，其中包含欧拉回路的不同 2 正规图的数量为（ ）。

- A.  $n!$
- B.  $(n-1)!$
- C.  $n!/2$
- D.  $(n-1)!/2$

显然  $n=3$  时答案只有是对的。



一种。代入算一下只有 D



## 第九题

---

正解：

容易发现所谓『有欧拉回路的 2-正规图』就是个简单环。考虑从 1 号点出发的路径，接下来按顺序经过的点方案共有  $(n-1)!$  种。每种对应一种连边方案。

但是从起点顺时针走一圈和逆时针走一圈是同一种连边方案，所以除以 2 去重。



## 第十题

共有 8 人选修了程序设计课程，期末大作业要求由 2 人组成的团队完成。假设不区分每个团队内 2 人的角色和作用，请问共有多少种可能的组队方案。（28）。

共有 8 个人，每个人可以选 7 个人组队。A 和 B 组队与 B 和 A 组队是同一种方案，故除以二去重。答案是  $7 \times 8 / 2 = 28$ 。

## 第 11 题

小明希望选到形如“省 A· $\mathcal{L}$   $\mathcal{L}$   $\mathcal{D}$   $\mathcal{D}$   $\mathcal{D}$ ”的车牌号。车牌号在“·”之前的内容固 的 5 位号码中，前 2 位必须是大写英文字母，后 3 位必须是阿拉伯数字（ $\mathcal{L}$  代表 A 至 Z， $\mathcal{D}$  表示 0 至 9，两个  $\mathcal{L}$  和三个  $\mathcal{D}$  之间可能相同也可能不同）。请问总共有多少个可供选择的车牌号。（676000）

方案数显然是  $26^2 \times 10^3$ 。

## 第 12 题

给定地址区间为 0~9 的哈希表，哈希函数为  $h(x) = x \% 10$ ，采用线性探查的冲突解决策略（对于出现冲突情况，会往后探查第一个空的地址存储；若地址 9 冲突了则从地址 0 重新开始探查）。哈希表初始为空表，依次存储(71, 23, 73, 99, 44, 79, 89)后，请问 89 存储在哈希表哪个地址中。 (2)

按顺序可以算出几个数存储的地址依次是 1,3,4,9,5,0,2。

## 第 13 题

对于给定的  $n$ ，分析以下代码段对应的时间复杂度，其中最为准确的时间复杂度为（ ）。

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

```
int i, j, k = 0;
for (i = 0; i < n; i++) {
    for (j = 1; j < n; j*=2) {
        k = k + n / 2;
    }
}
```

外层循环  $n$  轮，内层每轮运行  $\log n$  次。

## 第 14 题

以比较为基本运算，在  $n$  个数的数组中找最大的数，在最坏情况下至少要做（ ）次运算。

- A.  $n/2$
- B.  $n-1$
- C.  $n$
- D.  $n+1$

每次比较最多可以排除掉一个不是最大数的数据。所以无论如何都要比  $n-1$  次。

## 第 15 题

ack 函数在输入参数“(2,2)”时的返回值为 ( ) 。

- A. 5
- B. 7
- C. 9
- D. 13

```
unsigned ack(unsigned m, unsigned n) {  
    if (m == 0) return n + 1;  
    if (n == 0) return ack(m - 1, 1);  
    return ack(m - 1, ack(m, n - 1));  
}
```

手玩。可以画表辅助计算。

# 阅读程序 A



# 综述

---

- 这是一个字符串匹配算法。返回  $t$  在  $s$  中第一个出现的位置。
- `shift` 数组的作用是在  $t$  失配后跳过一段不可能重新匹配的位置。

## 判断题

---

16. 当输入为“abcde fg”是，输出为 -1(T)  
fg 不是 abcde 的子串。

17.当输入为“abbababbbab abab”时，输出为 4(F)  
注意到下标从 0 开始。输出应该是 3。

18.当输入为“GoodLuckCsp2022 22”时，第 20 行的“j++”语句执行次数为 2。(T)

理解程序以后很容易手玩。只会在最后 s 的最后两位进入 while 循环。

## 第 19 题

该算法最坏情况下的时间复杂度为 ( )。

- A.  $O(n + m)$
- B.  $O(n \log m)$
- C.  $O(m \log n)$
- D.  $O(nm)$

如果  $s = \text{aaaaaaaaa}\cdots$ ,  $t = \text{aaab}$ , 则  $i$  退化为每次  $+1$ , 每次都要跑满  $m-1$  次匹配。

## 第 20 题

---

$f(a,b)$  与下列 ( ) 语句的功能最为类似:

- A. `a.find(b)`
- B. `a.rfind(b)`
- C. `a.substr(b)`
- D. `a.compare(b)`

理解算法后易得。

## 第 21 题

---

当输入为“baaabaaabaaabaaaa aaaa”，第 20 行的“j++”语句执行次数为(10)。

手玩。

# 阅读程序 B

# 综述

---

这是一个以  $k$  为底的基数排序。

基数排序先对数字  $k$  进制下最底位进行排序，再对次低位进行排序……一直排到最高位。

代码中，`cnt[j]`（在做前缀和前）存的是当前位置有几个数的  $k$  进制第  $i$  位为  $j$ 。



## 判断题

---

22. 这是一个不稳定的排序算法 (F)

基数排序是稳定的。

23. 该算法的空间复杂度仅和  $n$  有关 (F)

和  $k$  也有关。

24. 该算法的时间复杂度为  $O(m(n + k))$  (T)

这里  $m$  其实是  $\log_k V$ 。容易看出这就是我们熟悉的基数排序复杂度  $O((n + k)\log_k V)$ 。

## 第 25 题

当输入为“5 3 98 26 91 37 46”时，程序第一次执行到第 36 行，`val[]` 数组的内容依次为（91 37 46 98 26）。

第一次执行时是对 3 进制最低位（`mod 3` 的值）进行排序。这几个数字 `mod 3` 分别是 2 2 1 1 1。故后三个数排序后在前面，前两个数排序后在后面。

关注到第 31 行是个倒序枚举，每次会把 `val[j]` 放到 `tmp[cnt[XXXXX]--]` 里。所以两个 `mod 3` 相同的数（他们 XXXXX 相同）`x,y`，如果 `y` 在后，会先把 `y` 放到 `tmp[cnt[XXXXX]]`，然后执行 `cnt[XXXXX]--`，`x` 在 `tmp` 里对应的下标比 `y` 小。所以在前面的数排完序后还是在前面。

## 第 26 题

若  $\text{val}[i]$  的最大值为 100,  $k$  取 ( ) 时算法运算次数最少。

- A. 2
- B. 3
- C. 10
- D. 不确定

程序时间复杂度是  $O((n + k)\log_k V)$ 。当  $n$  比  $k$  大很多的时候, 我们期望  $\log_k V$  尽量小, 此时  $k$  取 100 最优; 当  $n$  比  $k$  小很多时, 我们希望  $k$  尽量小。比如当  $n=2$  时,  $k$  取 10 显然比  $k$  取 100 要优。

## 第 27 题

当输入的  $k$  比  $val[i]$  的最大值还大时，该算法退化为（ ）算法。

- A. 选择排序
- B. 冒泡排序
- C. 计数排序
- D. 桶排序

计数排序：对每个数计算有多少个数小于它，这样就能找到它在排好序序列中的下标。

桶排序：把每个数字按某规则分配到桶里，在桶里分别排序（这一步有可能省略）后按顺序取出。

# 阅读程序 C

# 综述

---

- 这是一个诡异的类似进制转换的代码。但是第 20 行会改变  $n$  的正负号。所以它和一般的进制转换并不相同。



## 判断题

28. 该算法的时间复杂度为  $O(\log_k n)$ 。(T)

显然。

29. 删除第 23 行的强制类型转换，程序的行为不变(F)

删去强制类型转换后，因为 ans 是 int，加上 char 后依然会被转换成更高级的类型 int。所以输出的会是一些整数。

30. 除非输入的  $n$  为 0，否则程序输出的字符数为： $O(\lfloor \log_k |n| \rfloor + 1)$  (F)

这不是标准进制转换。



# 选择题

---

三个题目均手玩。答案为 ABB。

# 完善程序 A

## 综述

这是一个类似二分的过程。 $left1$ 、 $right1$ 、 $left2$ 、 $right2$  分别是还没有确定是否属于前  $k$  小的区间。

每次取两个区间的中点  $m1, m2$ ，计算两个前缀  $[0, m1]$  和  $[0, m2]$  的数字个数  $cnt$ 。如果  $cnt < k$ ，那么  $a1[m1]$  和  $a2[m2]$  较小的那个前缀一定是前  $k$  小的数。此时倍减那个区间（修改左端点）；否则较大的那个书后面的数都可以扔掉。此时修改右端点。

输出部分，在一个序列已经完全确定属于前  $k$  小的部分后，另一个序列属于前  $k$  小的部分自然也确定了。

## 选择题

DBCCA

34. 注意到下标是从 0 开始的，故要填  $m1 + 1 + m2 + 1$ 。
35. 这里再讨论是  $a1[m1]$  较小还是  $a2[m2]$  较小。结合 if 里的内容，可以看到 if 成立时，把 left1 左侧全选上，所以此时  $a1[m1]$  较小。
36. 运行到第 19 行时，必然两个区间有一个为空。这里判断的是哪个区间为空。结合下面  $left1 = 0$  时直接输出  $a2[k - 1]$  可以看出此时是第一个区间为空的情况。
37.  $a1$  取了 left 个数，故  $a2$  应该取  $k - left$  个数，下标从 0 开始，所以要再 -1。
38. 与 37 完全对称。

# 完善程序 B

## 解析

---

这是一个简单的记忆化搜索题。

$f[x][y]$  表示 1 当前有  $x$  升水，2 当前有  $y$  升水的答案。

分别讨论六种情况的转移。

## 选择题

ACAAC

39. 结合  $t = \min(a - x, y)$  可以看出这一情况是从 2 向 1 倒水。能倒出的水量是  $\min(1 \text{ 的剩余容量 } a-x, 2 \text{ 的当前水量 } y)$ 。所以状态转移到  $\text{dfs}(x + t, y - t) + 1$ 。

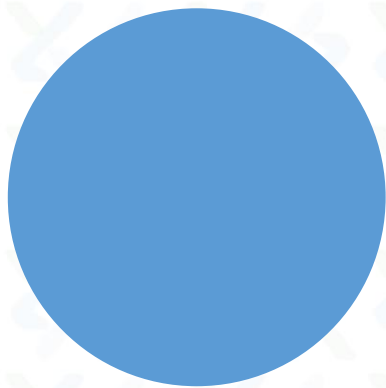
40. 与第 39 题完全对称。

41. 输出方案时，到达最终目标状态即为终点，无需再输出。

42. 与第 39 题完全对称

43. 与第 40 题完全对称。





End.

