



## 号爸信息交流总群...

群号：928983123



扫一扫二维码，加入群聊。



# 号家军 CSP-S 2021 初赛解析

---

## 一、单项选择题

1.

答案: A. ls

解析: 常识问题, 参考 NOI2021 笔试题库。

2.

答案: B. 01000000

解析: 计算可得, 注意进位。

3.

答案: A. 系统分配的栈空间溢出

解析: 常识问题。

4.

答案: C. 堆排序

解析: 稳定排序, 即不会交换相同元素。

插入排序: 每次找到最小值, 插入到前面它的排名处。如果找最小值时同时要求位置最小, 就不会交换相同元素。

冒泡排序: 每次遍历整个序列, 如果相邻元素大小关系不正确就交换, 一次遍历可以保证将最大值归位。

堆排序: 将所有元素插入到一个堆里, 再不断 pop。堆的结构不能保证不会交换相同元素。

归并排序: 每次将序列分为两半, 两边分别排好序之后再归并起来。只要保证归并遇到相同元素时, 先归并左边即可。

5.

答案: C.  $3n-2$

解析: 将  $2n$  个数分为  $n$  对, 每对之内先求出较大值和较小值, 再对  $n$  对中的较大值求 max, 对  $n$  对中的较小值求 min。

总比较次数为  $n + (n - 1) + (n - 1) = 3n - 2$ 。

6.

答案: C. 7

解析: 模拟即可。初始位置分别为 0, 1, 4, 9, 5, 3, 3, 5, 最后的 3 和 5 会被分别移动到 6 和 7。

7.

答案: C. 10

解析: 注意到留一个单点, 其余部分连成一个完全图最优。10 个点时的最大边数恰好为  $9(9 - 1)/2 = 36$ 。

8.

答案: B. 11

解析: 按题目中的定义, 高度为  $k$  的二叉树最多容纳  $2^k - 1$  个节点, 而  $2^{11} - 1 = 2047$ 。

9.

答案: D. 非叶子结点只有右子树的二叉树

解析: 设点  $u$  的左右儿子分别为  $l, r$ , 那么:

$u$  的先序遍历  $P(u) = u + P(l) + P(r)$ 。其中  $P(\emptyset) = \{\}$ , 加号表示序列的连接。

$u$  的中序遍历  $M(u) = M(l) + u + M(r)$ 。其中  $M(\emptyset) = \{\}$ , 加号表示序列的连接。

既然要求  $P(u) = M(u)$ , 那么容易发现  $l = \emptyset$ 。

10.

答案: A. 7

解析: 可以手动模拟, 也可以直接计算逆序对数。因为注意到一次交换消除一对逆序对是最优的。

11.

答案: A. 1

解析: 注意到  $\text{solve}(t, n) = 5^{t-1} \bmod n$ , 那么  $5^{22} \equiv 1 \pmod{23}$ 。这里可以手动计算但较为麻烦, 也可以使用费马小定理, 对于质数  $p$  和整数  $a$  ( $1 \leq a < p$ ), 有  $a^{p-1} \equiv 1 \pmod{p}$ 。

12.

答案: C.  $O(2^n)$

解析:  $F(1)$  和  $F(2)$  的调用次数是斐波那契级的, 四个选项中只有  $O(2^n)$  最接近。

13.

答案: C. 54

解析: 设  $f_i$  表示  $i$  个苹果, 不要求“选至少一个”的方案数。其中  $f_0 = 1, f_1 = 2, f_i = f_{i-1} + f_{i-2}$ 。那么答案就是  $f_8 - 1$ 。这道题也可以注意到只有 C 选项是“斐波那契减 1”的形式。

14.

答案: C. 165

解析: 先分别计算  $ab, bc, ac$  相等的情况, 再减去等边三角形的贡献。注意如果腰的长度是  $x$ , 那么底的长度必须小于  $2x$ 。这样可以算出钦定两条边为腰, 有 61 种方案。等边三角形有 9 个, 会被算三次, 所以答案为  $3 \times 61 - 2 \times 9 = 165$ 。

15.

答案: B. 19

解析: 注意到图是分层图, 且上一层的每个点到下一层的每个点都有边, 所以可以从后往前推。

## 二、阅读程序

下面判断题用 T 指代“对”，F 指代“错”。

(1)

16.

答案: T

解析: 对  $t$  的计算不涉及取整。

17.

答案: F

解析: 先 `/ sqrt(t)` 的话, 会被转化成 `double`; 但先 `/ 2` 的话, 是在 `int` 下进行的。前面的式子可能为奇数, 此时两个写法的结果不同。

18.

答案: F

解析: `sq` 的参数会被转为 `int`, 而  $x, y$  是 `double`。

19.

答案: T

解析: 注意到  $r = \pi/3$ 。然后就可以模拟了。

20.

答案: D. “4.1888”

解析: 依然是根据  $r = \pi/3$  模拟。

21.

答案: C. 求球的体积交

解析: 不认识公式也没关系。根据选项和程序中两个 `if` 的提示可以注意到,  $a, b, c$  是三维的坐标, 而  $d$  是球的半径。  $\sqrt{t}$  是两个球心的距离。程序中的两个 `if` 分别是“互相包含”和“互不接触”的情况。

(2)

注意到这个程序是在求最大子段和。 `Node` 结构体中:

- $h$ : 最大前缀和 (包括空前缀)
- $j$ : 最大子段和 (包括空段)
- $m$ : 最大后缀和 (包括空后缀)
- $w$ : 和

`solve1` 直接通过 `Node` 的合并求最大子段和。

`solve2` 是分治, 每层中计算了“经过中点的最大子段和”。

22.

答案: T

解析: 可以看到对  $n \leq 0$  的情况也做了判断, 而  $a$  的值不会导致 RE。solve1 和 solve2 都是在求最大子段和, 而对  $n \leq 0$  的判断又是一致的。

23.

答案: F

解析: 只有可能在  $n \leq 0$  的时候执行一次。

24.

答案: F

解析: 可以看到, 光 11 一个数就大于 7。

25.

答案: B.  $O(n)$

解析: 因为  $2^0 + 2^1 + \dots + 2^k = 2^{k+1} - 1$ 。所以 solve1 是  $O(n)$  的。也可以使用主定理计算  $T(n) = 2T(n/2) + O(1)$ 。

26.

答案: C.  $O(n \log n)$

解析: 可以看到 solve2(h,m) 调用 solve2(h,j) 和 solve2(j+1,m) 前进行了  $O(m-h)$  的计算。这是一个较为经典的复杂度计算。也可以使用主定理计算  $T(n) = 2T(n/2) + O(n)$  得到  $O(n \log n)$ 。

27.

答案: B. "17"

解析: 手动计算最大子段和即可。注意第一个数 10 是  $n$ , 不在序列内。一个快速的计算方法是, 先计算总和, 得到 14, 就可以排除 A. 13 和 D. 12。

(3)

这个程序是 base64 的加密和解密。

将输入的字符串的二进制写下, 每个字符是 8 位。

然后 6 位为一组重新组合, 在大小为  $2^6 = 64$  的表中查询, 得到加密后的字符串。

encode 函数中三个字符为一组的原因是  $3 \times 8 = 4 \times 6$ 。

28.

答案: F

解析: 解压出来的字符串可能带有 `\n` 字符, 所以不一定是“一行”。

29.

答案: T

解析: 由加密解密的过程可得。

30.

答案: F

解析: 计算可得。

31.

答案: B.  $O(n)$

解析: 观察程序可得。encode 中只有一个  $O(n)$  的 for，里面只有简单运算和 string 的 += 操作。

32.

答案: B. "255" / D. "-1" (有争议)

解析: 可以看到第一行是输出了 `int(char(0xff))`。这里的问题就是 char 是有符号还是无符号。如果有符号就会输出 -1，否则就会输出 255，据说这里是“实现定义的”。

33.

答案: D. "Q1NQMJyMWNzcA=="

解析: 首先注意到 CSP2021csp 的长度模 3 等于 1，也就是串的末尾应当有两个 =，因此排除 A 和 C。

然后就是一个 g 和 w 的问题了，这里只有一位，可以通过计算来判断。

### 三、完善程序

(1)

可以看到是使用了类似 dijkstra 的算法。F[i] 表示“表示出 i 至少需要多少个 4”，而 vis[i] 表示“F[i] 是否已经确定”。

34.

答案: D.  $F[4] = 1$

解析: 这是 dijkstra 的边界条件。

35.

答案: A. !Vis[n]

解析: 这是由 dijkstra 中，vis 的定义得到的。

36.

答案: D. !Vis[i] && F[i] < F[x]

解析: 这里是 dijkstra 中，每次选择确定的数中 F 最小的数的过程，但这里用了  $O(M)$  的循环。

37.

答案: C. Vis[i]

解析: 这个 if 其实可以去掉，但在四个选项中，只有 C 是正确的，这可以由 vis 的定义得到。

## (2)

这个程序整体的过程很清晰，先处理笛卡尔树，再 dfs 求欧拉序，再求块之间的最小值，再求块内的最小值。

$O(n)$  求笛卡尔树的过程：

用增量法，每次在当前序列右侧加入一个数。

注意到加入的时候，只可能影响到从根向右延伸的一条链。即，从这条链的某个位置断开，将下面的部分作为当前点的左儿子，然后将当前点接在断开位置的右儿子。

所以可以用栈来维护这条链，每次弹栈弹到需要的位置即可。

38.

答案：A.  $p \rightarrow \text{son}[0] = S[\text{top}--]$

解析：对应建笛卡尔树时，“将链接在当前点的左儿子”的部分。

39.

答案：D.  $S[\text{top}] \rightarrow \text{son}[1] = p$

解析：对应建笛卡尔树时，“将当前点接在断开位置的右儿子”的部分。

40.

答案：A.  $x \rightarrow \text{dep} < y \rightarrow \text{dep}$

解析：注意这里是在 dfs 树上求 lca，也就是深度越浅越优。

41.

答案：D.  $A[i * b + j] \rightarrow \text{dep} < A[i * b + j - 1] \rightarrow \text{dep}$

解析：这里只有这一个选项是根据 dep 来比较的。从这里也可以看出，在 mask 中的第  $i - 1$  位是 1，代表  $a[i] = a[i - 1] - 1$ ，这个信息可以用在第 41 题。

42

答案：D.  $v += (S \gg (i - 1) \& 1) ? -1 : 1$

这里就是用到了第 41 题中提供的信息。

43.

答案：C.  $(\text{Diff}[p] \gg (l - p * b)) \& ((1 \ll (r - l)) - 1)$

解析：从第 89 行的 `A[1 + Pos[S]]` 可以看出，在 mask 中，较低位是靠“左”的，所以要右移的内容和  $l$  有关。