

提高级 CSP-S 第 3 套初赛模拟试题答案及解析

一、单项选择题

题号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
答案	B	C	B	A	B	D	A	A	B	C	A	B	A	B	C

【解析】

1. a 是数组常量, 不能赋值。
2. A 是华为的 CPU, B、D 分别为 Intel、AMD 公司产的 CPU, C 是显卡。
3. 多种解题方式。将两个数转换为 10 进制数相加得 3059, 然后分别转换为二进制、八进制数、十六进制进行比较, 选 B。

4. 原理如下:

先序遍历顺序是: Mid-Left-Right;

后序遍历顺序是: Left-Right-Mid;

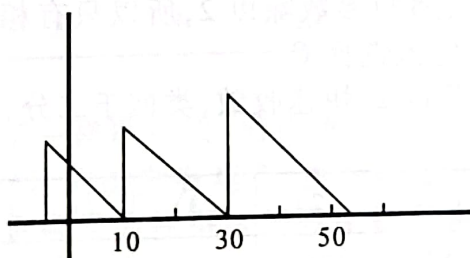
可以看到, 只有中间的结点 (M) 顺序变化了, 左右结点相对位置是不变的。那可以推断出, 要满足题意“二叉树的先序遍历与后序遍历正好相反”, 说明整个二叉树左子树或者右子树有一个没有 (遍历就成了, 先: M-L, 后: L-M; 或者先: M-R, 后: R-M) 也就是必然是一条链。

所以只有 A 对了。

5. 字符串后有一个结束标记“\0”也占一个字节, 故 x 是 6 个字节, y 是 5 个字节。

6. 如下图所示, 三个三角形面积除以 60 即为期望。

$$\text{即 } \frac{\left(\frac{15^2}{2} + \frac{20^2}{2} + \frac{25^2}{2} \right)}{60} = 10 \text{ 分 } 25 \text{ 秒}$$



7. 该题可以直接针对选项去排除, 第一个元素是 Q , 看选项中 Q 之前的元素是否均小于 Q , 之后的元素是否都大于 Q 。如果仅有一个选项满足, 则可以选择, 如果有多个满足这样条件的选项, 那么再按快排的方法, 看看其他元素位置是否正确。

8. n 个点要组成一棵树, 应该是 $n-1$ 条边, 现有 m 条边, 故需要删除 $m-n+1$ 条边。
9. 先放蓝球和白球, 有 P_{10}^2 种方法, 再放红球有 C_8^2 种方法, 故共有 $P_{10}^2 C_8^2$ 种放法。
10. 由于单位木材桌子价格更高, 所以优先考虑生产桌子, 生产 5 张桌子的话, 剩余木材不够生产任何成品, 可以卖 150 元, 这个时候要合理利用剩余木材, 可以少生产 1 张桌子, 从而使得木材可以生产 2 张椅子, 故最终可以卖出 160 元。
11. 插入排序、冒泡排序、选择排序复杂程度均为 $O(n^2)$, 快速排序复杂程度为 $O(n \log n)$, 桶排序复杂度为 $O(n)$ 。
12. 二叉树是树形结构, 不属于线性结构。
13. Dijkstra 算法只适用无负权值的单源点最短路。Floyd 算法可以处理多源点, 适用有负权值, 但复杂度高。Bellman-ford 算法是求含负权图的单源最短路径的一种算法。SPFA 可以处理负边权, 但不能处理负权回路。
14. 栈的特性: 对于取出栈内元素每次只能从栈顶开始取[后进先出(栈满时, 只能先出后进)]。
由于栈内只能容纳 4 个元素: 所以 5, 6 不可能第一个出栈; 当栈内少于四个元素时, 既可以选择进栈, 也可以选择出栈; 所以 1, 2, 3, 4 中第一个出栈的元素是随机的, 主要是从第二个开始看。
本题考查逻辑思考和判断的能力。

15. 假设 3 条平行线 a, b, c 上分别有 7, 5, 6 个点, 其中 a 上 7 个点中任取 2 点有 $\frac{7 * (7-1)}{2} =$

21 种取法;

同理可知 b 上 5 个点中任取 2 点有 10 种取法; c 上 6 个点中任取 2 点有 15 种取法。

a 上任取 2 点, b 上任取 2 点, 可以组成的四边形个数是 $21 * 10 = 210$;

a 上任取 2 点, c 上任取 2 点, 可以组成的四边形个数是 $21 * 15 = 315$;

b 上任取 2 点, c 上任取 2 点, 可以组成的四边形个数是 $10 * 15 = 150$;

a 上任取 1 点, b 上任取 1 点, c 上取 2 点, 可以组成四边形个数是 $7 * 5 * 15 = 525$
(不同直线上三个点都不在同一条直线上, 所以肯定可以组成四边形);

a 上任取 1 点, b 上任取 2 点, c 上取 1 点, 可以组成四边形个数是 $7 * 10 * 6 = 420$;

a 上任取 2 点, b 上任取 1 点, c 上取 1 点, 可以组成四边形个数是 $21 * 5 * 6 = 630$ 。

所以可以组成四边形个数是 $210 + 315 + 150 + 525 + 420 + 630 = 2250$ 。

二、阅读程序

1.

题号	(1)	(2)	(3)	(4)	(5)	(6)
答案	√	×	×	√	C	B

【解析】

(1) 递归结束边界改变, 但这 2 个边界对于正整数 n 来说, 值相同。

(2) 输出的结果每进行一次递归, 值会变大, 很显然 n 越大, 值远远超过 n 。

(3) 递归每次调用, 参数值除以 2, 所以负数最终也能满足递归到 0 的结束条件。

(4) 由于每次递归调用时均会乘以 n , 所以随着 n 的增加, 输出的结果会越来越大。

(5) 因为输入的整数 n , 每次递归参数除以 2, 所以只有相差 1 的负数才有可能符号不一样, 每次在 2^k 处变符号, 故应该选择 C。

(6) 该程序每次调用, 参数除以 2, 快速收敛, 类似于二分, 复杂程度应该为 $\log n$ 。

2.

题号	(1)	(2)	(3)	(4)	(5)	(6)
答案	√	×	×	√	A	A

【解析】

(1) 字符数组应该有一个字符串结束的标志“\0”。

- (2) 如果输入的是从小到大的, 才会输出 “No result!”。
- (3) 18 行到 22 行实际是要找出从 i 位置到最后比 $str[i-1]$ 大的最小的字符。
- (4) $chr[]$ 数组就相当于桶, 按照从小到大存储了出现的字符的个数 (不包括前面已经输出的字符)。
- (5) 这行代码是输出从后开始数第一个正序字符 (即前一个字符比后一个小) 之前的字母, 故应为 A。
- (6) 根据题意, 应该从后往前, 先找到第一个前面比后面小的字符, 然后前面的字符一一输出, 之后找比出现字符大的最小值输出, 然后针对 4 个选项一一排查。

3.

题号	(1)	(2)	(3)	(4)	(5)
答案	✓	×	✓	×	A

【解析】

- (1) 由 46~47 行确定蛇身具体位置。
- (2) `check` 函数显然是用来检测蛇是否因越界致死。
- (3) 简单模拟如 58~71 行所示。
- (4) 本题本质就是个模拟题, 只要一步一步代进程序规划即可得解, 可表示为: 贪吃蛇在第 9 秒后就死亡了, 因此最后贪吃蛇保持的是死亡前 (第 8 秒过后)。
- (5) 这道题主要的时间都是用在模拟贪吃蛇的运动以及最后的绘图之上, 很容易得到最后的渐进时间复杂度为 $O(x^2)$ 。

三、完善程序

1.

题号	(1)	(2)	(3)	(4)	(5)
答案	B	C	A	D	A

【解析】

- (1) 找最短路径, 发现有一条到 $[dx][dy]$ 点更短的路, 因为走一格需要 1 个步长, 所以要 +1。
- (2) 很简单, 发现有更短的路就更新。
- (3) 发现更短路径后更新该点路径, 还需要加入队列, 从而可以松弛更新后续结点的路径值。
- (4) 搜索时的第一个参数为 0, 是以小 D 所在位置作为源点进行搜索, 第一个参数为 1, 则以宿舍作为源点进行搜索。
- (5) 很显然, 我们需要的是从小 D 位置到该 KFC 所在点及从宿舍位置到该 KFC 所在点位置距离和的最小值, 故应该选择 A。

2.

题号	(1)	(2)	(3)	(4)	(5)
答案	A	B	A	D	D

【解析】

尺取法: 顾名思义, 像尺子一样取一段, 借用挑战书上面的话说, 尺取法通常是对数组保存一对下标, 即所选取的区间的左右端点, 然后根据实际情况不断地推进区间左右端点以得出答案。尺取法比直接暴力枚举区间效率高很多, 尤其是数据量大的时候, 所以说尺取法是一种高效的枚举区间的方法, 是一种技巧, 一般用于求取有一定限制的区间个数或最短的区间等。当然, 任何技巧都存在其不足的地方, 有些情况下尺取法不可行, 无法得出正确答案, 所以要先判断是否可以使用尺取法, 再进行计算。

使用尺取法时应清楚以下四点：

- ①什么情况下能使用尺取法？②何时推进区间的端点？③如何推进区间的端点？④何时结束区间的枚举？

尺取法通常适用于选取区间有一定规律，或者说所选取的区间有一定的变化趋势的情况，通俗地说，在对所选取区间进行判断之后，我们可以明确如何进一步有方向地推进区间端点以求解满足条件的区间，如果已经判断了目前所选取的区间，却无法确定所要求解的区间如何进一步得到根据其端点得到，那么尺取法便是不可行的。首先，明确题目所要求解的量之后，区间左右端点一般从整个数组的最起点开始，之后判断区间是否符合条件，再根据实际情况变化区间的端点求解答案。