

提高组 CSP-S 2025 年初赛模拟卷 1 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	A	C	C	D	B	C	A	C	B	A	B	B	D	C

【解析】

1. 【数学：集合、数论】逐个代入，看哪个数只有因子 2 和 3，则该数必然被包含。
2. 【STL 知识】`top()` 是 STL `stack` 里面的函数，不是 `vector` 里面的函数。
3. 【Linux 命令】只有 `ls` 是 Linux 中的命令，`dir` 是 Windows 系统中的命令。`show` 和 `display` 一般是交换机里面的命令。
4. 【二进制位运算】-5 和 -6 为有符号整数，计算机里面一般用补码表示。负数补码是通过将绝对值的二进制表示取反，并加 1 得到的。-5 的二进制补码表示为 11111011，-6 的二进制补码表示为 11111010，两者异或得到 1，正整数的补码和原码相同，所以原码也为 1。
5. 【数据结构：树】设每棵树的点分别为 $a_1, a_2, \dots, a_y$ 。那么每棵树的边分别为 $a_1-1, a_2-1, \dots, a_y-1$ 。由题意可知 $a_1+a_2+\dots+a_y=x$ ，森林的总边数为每棵树的边数之和。 $\text{Sum} = (a_1-1)+(a_2-1)+\dots+(a_y-1)=x-y$ 。
6. 【排序算法：拓扑排序】拓扑排序要解决的问题是一个有向无环图的所有结点排序，适合这种有先后依赖关系的排课。其他的选项一般用在常规的排序中，比如将整数数组从小到大排序场景。
7. 迪杰斯特拉算法在最坏情况下的时间复杂度为 $O(n^2)$ 。
8. FIFO (First In First Out, 先进先出) 的数据结构是队列 (queue)。
9. 【类：类的概念】对于成员数据的访问，修改和调用均需要通过 `public` 的成员函数来进行。
10. 每层一个元素，共 2025 层。
11. 哈夫曼算法中不能有重复的子串。
12. 间接递归层次过深可能会引发栈溢出的错误。
13. $P(A) = P(B)$, $P(A \cup B) = P(A) + P(B) - P(A)P(B) = 0.64$ ，解得 $P(A) = 0.4$ 。
14. 先确定拿三本书的人，有 5 种选法，再在 8 本书中为他分配书，有 $C(8, 3)=56$ 种选法；再确定拿两本书的人，有 4 种选法，再在 5 本书中为他分配书，有 $C(5, 2)=10$

种选法；此时剩下三个人对应三本书，有 $A(3,3)=6$ 种选法。因此所求种数为 $5 \times 56 \times 4 \times 10 \times 6 = 67200$ 种。

15. 快速排序算法的第一轮只能保证选择的那一个基数在正确位置，其他都不能保证。

二、阅读程序

(1)

题号	16	17	18	19	20	21
答案	×	×	√	×	C	B

【解析】

第 16 题：一个数位 dp，题目求的是 $[l, r]$ 区间内二进制下 0 的数目大于等于 1 的数目的个数，很板子。

16. 该程序能够计算 $[l, r]$ 区间内所有二进制下有且仅有一个 0 的数字（例如 10, 101, 1101）。

17. 该程序获得结果 1。

18. 该程序获得结果 6。

19. 该程序 solve 会执行两次，因此最多会执行 $2 \times \log(1e15)$ 次。

20. A. 时间复杂度为 $O(\log N^2)$ ；

B. 根据 IEEE754, double 的 M 位有 52 位，因此能够无损转化到 $2^{52}-1$ 以内的数字，a 的大小只有 50，因此不能；

C. 二进制下全 1，因此答案为 0；

D. 同 B 选项，数组开小了。

21. 直接用二进制表示 1~14 的数，发现符合题意的区间是 $[2, 12]$ ，答案为 6，其他选项为 5。

(2)

题号	22	23	24	25	26
答案	×	√	√	D	B

【解析】

把 breeds 中的东西看成物品，ExtendKnapsack 函数实现了一个背包的扩张，Knapsack 表示装到价值位 total 至少需要多少个物品。

22. 一开始只有 $\text{knapsack}[0]=0$ 。

23. 根据考纲出的一个题，就是抛出异常。
24. 如果 0 是正确答案并且 b 不全为 0，说明没东西装进背包，但是没装进背包就不会输出用了大于 0 个物品。
25. 直接模拟一次，第二个装进一个 4，第三个装进一个 2，第四个装进一个 2 和一个 1，用了四个。
26. A. 计 $bmax=K$, breeds 的 $size=m$ ，那么复杂度为 $O(MK)$ ，两个复杂度一样。
 B. 本来就是实现一个完全背包。
 C. N 应该指的是 breeds 物品数量。
 D. 例如 $n=3, m=1, breed=\{1\}, volumes=1, 1, 1$ ，实际输出 3。

(3)

题号	27	28	29	30	31	32
答案	×	×	D	C	C	B

【解析】

`dp[i][j].first`: 前 i 个字符中匹配到 `bessie` 第 j 位时的最大完整出现次数。

`dp[i][j].second`: 在达到上述次数的前提下，删除字符的最小代价总和。

27. 改为出现次数最多时最小的删除代价和。
28. 里面循环的范围也需要更改。
29. 见上方解析。
30. 匹配到一次完整的 `bessie`，代价和为 0。
31. 本题数组开的是两百万，四个选项都不会炸内存，C 选项炸 `long long` 了。
32. A. 根据代码，当 `first` 不相同，返回 `first` 较大的那个，因此应该返回 `{2, 3}`。
 B. 没问题。
 C. 不可能，`ans` 不可能是负数。
 D. 这段代码的时间复杂度为 $O(N)$ 。

三、完善程序

(1)

题号	33	34	35	36	37
答案	B	A	C	A	B

【解析】

显然可以用贪心的思路来做这道题。

假设当前在第 i 个加油站，汽车现有油量为 $curGas$ ，到达下一个花费更低的加油站需要的油量为 $gasNeed$ ，那么有以下几种情况：

- 如果 $curGas \geq gasNeed$ ，则直接过去，去那边再加油。
- 如果 $curGas < gasNeed$ ，则先在这个加油站加油到刚好够去下一个加油站，再过去。
- 如果之后没有价格更低的加油站，则先把油加满。

在本题中 $nextSmall[i]$ 表示 i 的下一个费用比它低的加油站，用一个栈完成。注意这里是先放进去再++的，因此下标是从 0 开始的。

如果当前油量到不了下一个加油站的话，就只能输出 -1 了。（没考查这一点，不用管）

33. 这就是个排序的操作。

34. 这里 s 的下标从 0 开始。

35. 见上面的解析，题目明确了 $=0$ 也是合法情况。

36. 应该到下一个比它价格低的加油站位置。

37. 在这个加油站加需要的油-已经有的油。

(2)

题号	38	39	40	41	42
答案	C	D	B	D	B

【解析】

分析，这是一个换根 dp，dfs1 求得从 1 为根到叶子结点的文件路径长度。

接下来就是第二个 dfs 来做换根操作。如果 x 换成根，那么它的父亲都会变成它的子文件。假设 x 的儿子是 y ，如果想把根从 x 转化为 y ，那么：

首先， y 的所有文件都会从 $y\backslash***$ 变成 $***$ ，这里路径长度会减少 $strlen(y)+1$ 。其次，其他文件都会从 $***$ 变成 $..\backslash***$ ，这里路径长度会增加 3。

38. $(child \rightarrow namelen + (child \rightarrow isFile ? 0 : 1))$ 是路径长度增量，考虑文件夹需要加斜杠。

39. 一个点到所有文件的相对距离 = 第一次 dfs 得到它自己的子树 + 第二次 dfs 从父亲处得到的。

40. 从前面的解析可以得到结果。

41. 见第 58 行注释，如果是文件，则 $numChildren=0$ ，只有 D 选项满足。

42. 就是找到总路径最小的，因此选 B。