

提高组 CSP-S 2025 年初赛模拟卷 5 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	D	A	C	D	C	A	C	C	D	B	C	B	A	B	D

【解析】

- 析构函数是一个特殊函数，在对象生命周期结束时被自动调用，执行清理任务。
- 文件操作 `rm -r` 是删除文件夹，很难恢复，需要谨慎。
- `front` 是 `deque` 里面的操作函数，不属于 `set`（集合）。
- 代入计算即可，逻辑表达式的计算问题。
- 0-1 背包问题，直接模拟计算可以得出结果 110。
- 根结点是 1，452 和 367 分别在左子树和右子树中，不断地推理可知 A 选项正确。
- 二叉树释放 n 个结点的算法时间复杂度是 $O(n)$ ，每个结点仅遍历一次。
- 动态规划算法有关的题目。自动机是处理字符串的算法。
- 矩阵除了加减运算，也可以做乘法运算。
- $n^5 - 5n^3 + 4n + 15 = n(n^2 - 1)(n^2 - 4) + 15 = (n - 2)(n - 1)n(n + 1)(n + 2) + 15$ ，显然 $(n - 2)(n - 1)n(n + 1)(n + 2) \% 4 = 0$ ，从而 $(n^5 - 5n^3 + 4n + 15) \% 4 = 15 \% 4 = 3$ 。众所周知，完全平方数 $\% 4 = 0$ 或者 1，所以答案是 0 个。
- 双向队列是线性数据结构。其他三种都是树形数据结构。
- 题目中只有插入排序、归并排序、冒泡排序和桶排序是稳定的。
- 数论题。 $n + 1 = a + b + a * b + 1 = (a + 1)(b + 1)$ ，只要 $n + 1$ 是合数即可， a, b 最小从 1 开始， $n + 1$ 从 4 开始到 100，共有 74 个合数，所以也有 74 个奇特征数。
- 全部情形共有 $7!$ 种。记第 i 把锁被打开的情形构成集合 A_i ， $i = 1, 2, 3$ 。则 $|A_i| = 6!$ ， $|A_i \cap A_j| = 5!$ ， $|A_1 \cap A_2 \cap A_3| = 4!$ ，由容斥原理可知概率为 $(7! - 3 * 6! + 3 * 5! - 4!) / 7! = 67 / 105$ 。
- 复杂哈希函数是避免冲突发生，不是解决冲突。

二、阅读程序

(1)

题号	16	17	18	19	20
答案	×	√	√	C	B

【解析】

本段代码主要考查时空复杂度与读题，以及第 53 行的 if 语句。

16. 代码中有一个 floyd，点的数量取决于 1 的个数 t ，即时间复杂度为 $O(t^3)$ 。

17. f 数组使用了 $O(n^2)$ 的空间。

18. 所有位置均为 1 时 $T=n*m$ 。

19. $f[i][j]$ 为 1，说明第 i 个 1 与第 j 个 1 相邻（且 $i < j$ ），直接相邻的对数为行内左右相邻 8 对（行 2 和行 4 各 4 对）和上下相邻 4 对（列 3 的垂直链），总计 12 对。

20. 直接相邻的对包括行内左右 6 对（行 3 的 2 对，行 4 的 4 对）和上下相邻 8 对，总计 14 对。

(2)

题号	21	22	23	24	25	26	27
答案	×	×	×	×	A	C	C

【解析】

本段代码不难，因此在考查读代码本身的基础上，考查了一些 std 函数以及其他细节。

21. `less<int>` 表示让数组从小到大排序。

22. `std::sort()` 的时间复杂度并不是随机的，而是 $n \log n$ 。

23. 快速排序不是一种稳定的排序算法。

24. 这里 `+` 的优先级高于位运算，因此去掉括号变成了 `n >> (1+1)`。

25. 可以推出：

`a = [5, 5, 8, 10, 13, 14, 19]`

`b = [25, 18, 17, 14, 4, 3, 2]`

`f[1] = 5 ('f')`

`f[2] = 25 ('z')`

`f[3] = 5 ('f')`

`f[4] = 18 ('s')`

`f[5] = 8 ('i')`

`f[6] = 17 ('r')`

`f[7] = 10 ('k')`

26. A. `vector size` 实现的时间复杂度是 $O(1)$;
 B. 考虑 `abcdef`, 程序会输出 `afbecd`;
 C. 例如 `a` 和 `b`, 必然输出其中一个;
 D. 显然不正确。
27. 存在一种情况, `pos=0`, 此时仅需要看 `1~100` 中有多少个奇数, 显然有 `50` 个。

(3)

题号	28	29	30	31	32
答案	√	D	D	A	A

【解析】

本题是一个线段树优化 dp, 主要考查线段树相关的性质, 通过阅读代码寻找状态转移方程。

28. 线段树需要四倍的空间。
29. 根据第 58~63 行可以得知 `i` 的取值每次加 2, 且我们查询的区间是 $[i-2*b, i-2*a]$ 。然后 `dp[i] = min_{该区间} (dp[j])`。
30. 根据转移方程直接计算, 不需要带入线段树。
31. 由于存在区间 $[1, 8]$, 此时无法转移到 `dp[l]`, 因此输出 `-1`。
32. A. 根据差分可得。
 B. 不可以, 线段树结点的下标从 1 开始。
 C. 每个线段的半径是一个整数, 所以覆盖的长度一定是偶数, 让所有地方都被覆盖到, 奇数的坐标自然不能作为端点。
 D. 不可以, 线段树结点的下标从 1 开始。

三、完善程序

(1)

题号	33	34	35	36	37
答案	A	C	D	C	B

【解析】

因为 $(1, 1)$ 只能被一个矩阵影响, 所以如果要把 $(1, 1)$ 改成 0, 就必须让这个子矩阵的所有数都改变。同理, $(1, 1)$ 处理后, $(1, 2)$ 只能被一个矩阵影响, 以此类推, 用差

分来维护就可以了。

33. 这里判定边界, 因为是 $i+k, j+k$, 所以只需要判断大于 n 。

34. 二位差分需要选择四个区域:

```
d[x][y] -= c;
```

```
d[xx][y] += c;
```

```
d[x][yy] += c;
```

```
d[xx][yy] -= c;
```

仅有第四步没有做。

35. $val[i][j] = d[i][j] + d[i-1][j] + d[i][j-1] - d[i-1][j-1]$ 。

36. $f[i][j]$ 是初始值。

37. 同 36 题, bb 要进行方向操作, 将值减小 1, 将次数 bs 增加 1。

(2)

题号	38	39	40	41	42
答案	A	B	B	A	C

【解析】

38. 这里应填 $\text{pow}(2, i)$, 但是没有这个选项, 又因为 i 最大可以是 62, 因此选等价的 $1ll < i$ 。

39. 字典树插入的时候更新, 根据程序分析可以得到结果。

40. 要取异或和最大, 肯定每次找相反的点。

41. P 数组是中间过渡数组, 是用来承接 $i-1$ 的数值, 帮助更新 $c[i]$ 的, 根据 c 的转移方程可知应选 A。

42. 根据题意, 每次删除 $bk[i]$ 后再查询。