

提高组 CSP-S 2025 年初赛模拟卷 3 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	B	B	A	B	C	C	A	D	B	C	C	A	B	C

【解析】

- 典型的欧拉筛算法，又叫线性筛算法。
- `vi` 是编辑命令，`more` 是分页显示，`cat` 是普通显示，`ls` 是显示文件和文件夹。
- JPEG 是图片格式，MPEG、AVI 和 WMV 是视频格式。
- C++ 中左值必须是可以修改的，显然 A 选项错误，B 选项中的 `+q` 代表 `q` 的正值。
- 对于 B 选项，可以举反例， $n=1$ 时边为 0。
- 根据前、中、后序遍历的定义，很容易画图找出反例，只有 C 选项正确。
- `exe` 和 `bat` 是 Windows 下的可执行可批处理文件，只有 C 选项正确。
- 分类讨论。一种是 3 个奇数，3 和 9 不能同时选，共有 7 种。一种是两个奇数加一个偶数，暴力枚举有 35 种。共 $7+35=42$ 种。
- 快速排序在最坏情况下的时间复杂度是 $O(n^2)$ 。
- KMP 是字符串的算法，不属于贪心算法。
- 显然 C 选项不可能，因为深度优先搜索 8 直接到不了 6。
- 两个循环中第一个循环的时间复杂度是 $O(n)$ ，主要看第二个双重循环，因为第二重是 `j+=i`，这是调和级数，所以时间复杂度为 $O(n\log n)$ 。
- 递推显然比递归效率要高，A 选项说反了。
- 首先可以从 50 支队伍中找出 24 支战胜过外部所有队伍的，记为 $A_1, A_2, \dots, A_{24}$ ，同时也可以找出 24 支全负于外部所有队伍的，记为 $B_1, B_2, \dots, B_{24}$ ，下面说明这两组队伍组内具有严格的胜负关系（即不存在环）。
若存在环 A_1 胜 A_2 胜 A_3 胜 $\dots$ 胜 A_k 胜 A_1 ($K \leq 24$)，则取 $A_1, A_2, \dots, A_k, B_1, B_2, \dots, B_{27-k}$ 这些队伍，其中没有全胜队，从而 A 组内的胜负关系必然是严格单调的。类似地，B 组也完全一样。
从而 A 组中所有球队的积分是 $49, 48, \dots, 26$ ，B 组中所有球队的积分是 $0, 1, 2, \dots, 23$ ，那么中间两支球队只能是 24 分和 25 分。总共 50 种分数。
- 最坏情况就是 4 个点全连通用了 6 条边，还需要接入 5 个点并连通起来，至少还需要 5 条边。

二、阅读程序

(1)

题号	16	17	18	19	20	21
答案	√	×	×	×	C	C

【解析】

16. `std::endl` 会触发 `flush`，所以大量使用时往往比单纯 `'\n'` 要慢。
17. `priority_queue` 定义在 `<queue>` 里，而不是 `<algorithm>`。
18. 这里应该是按照 `w` 从大到小排序。
19. 这里交换后逻辑不一样，交换后等于情况处理不了。
20. 排序后实际存放的是 $(9,4), (7,6), (5,5), (9,4), (7,6), (5,5), (9,4), (7,6), (5,5)$ ，但外层 `for` 是从最小 `www` 向大处理，即先处理 $(5,5), (5,5), (5,5)$ ，再处理 $(7,6), (7,6), (7,6)$ ，最后处理 $(9,4), (9,4), (9,4)$ 。
21. 一次 `sort(a+1, a+n+1)`，总的 `push` 和 `pop` 也是 $O(N)$ 级别，单次的时间复杂度是 $O(\log N)$ 。

(2)

题号	22	23	24	25	26	27
答案	√	√	√	A	B	D

【解析】

本题是一个 `kruskal` 重构树的题，按照其思路，两者之间最大 `val` 的最小值会在 `lca` 中，因此使用数链剖分求 `LCA`。需要注意的是，树中最多有 $2n-1$ 个结点（当且仅当图连通时）。

22. 程序里确实是先 `dfs1` 找重儿子、计算子树大小 `tot[u]`，再 `dfs2` 处理 `top[u]`，从而实现 `LCA` 查询。
23. 每次两个结点合并后会增加一个结点，因此最后会有 $2N-1$ 个结点。
24. 如果可以撤销的话，就不能使用路径压缩。
25. 第一个不连通，输出 `-1`；第二个走 `2->1->4`，输出 `3`；第三个走 `1->2->3`，输出 `5`。
26. 其他地方也是 $n \log n$ 的，这里对复杂度没有影响。
27. `D` 选项：如果图是连通的，这里显然只会进行一次，但是这里并不保证图连通。

(3)

题号	28	29	30	31	32	33
答案	√	×	√	√	C	D

【解析】

28. lowbit 的性质是找到最后一个 1 以及后面的 0 对应的数字。
29. 这里 $n+1$ 是离散化后的最大值, 如果设为 $1e6+1$, 那么树状数组也要相应地开大。最后一个循环会访问这个位置。
30. L, R 分别代表左端最长不上升序列的长度和右端最长不下降序列的长度, 则 val 单调不降时就有 $L[i]+R[i]=n+1$, 可能比答案大。
31. 排序和树状数组的复杂度都是 $O(n\log n)$ 。
32. 模拟可得 4。
33. 选项 A 错误, 值域不变, 只需要开大 10 倍。选项 B 错误, 总和为 $O(n)$ 。选项 C 错误, ans 最大为 n 。

三、完善程序

(1)

题号	34	35	36	37	38
答案	A	C	D	D	B

【解析】

这是一种差分的做法, 变量名提示得很明显。

34. 我们需要将差分数组中的值累加到 v 上, 以计算出每个结点的实际点权。
35. 创建一个无向图, 使得每个结点都能记录与之相连的其他结点。
36. `depdfs(1, 0, 1);`。因为通常我们将树的根结点定为 1。
37. 需要增加从 $u[i]$ 出发能到达的所有结点的点权, 通过在根结点加上 val , 并在 $v[i]$ 处减去 val 来实现差分。
38. 我们需要增加从 $v[i]$ 出发能到达的所有结点的点权。

(2)

题号	39	40	41	42	43
答案	C	C	D	D	B

【解析】

同一列的两个人最多会被交换一次，所以对于每一列，其实只有两种选择：换或者不换。对于两个相同的数字，如果它们在同一行中，那么对于它们各自所在的列而言，这两列中必须一个换而另一个不换；如果它们不在同一行中，那么两列要么都不换，要么一起换。每一列当作一个结点的话，那么对于任意两列，如果它们之间存在上述关系中的某一种，就把这两列所对应的结点连边。

对每个连通块进行黑白染色，输出颜色数量的最小值。将每个连通块中的最少的黑色结点个数相加，即可得到答案。

39. 如果被访问两次，就是 `vis2` 被标记，一次被 `vis1` 标记，0 次不被标记。

40. 因为函数返回的是一个 `pair`，所以肯定选 C。

41. 根据上面的意思，就是染色，统计当前某个颜色的点的数量。

42. 给下一个点染色，肯定只与当前点和下一个点与两者之间的边有关。

43. 如果每行各出现一次，两者可以一起换，建立 `type=0` 的边。