洛谷网校

2025 LUOGU 非专业级别收容能力认证第一轮

（SCP-S1）提高级 C++语言试题

认证时间：2025 年 8 月 14 日 14:30~16:30

关注 洛谷科技 公众号

考生注意事项：

- 试题纸共有 17 页，满分 100 分。请在洛谷作答，写在试题纸上的一律无效。
- 不得使用任何电子设备（如计算器、手机、电子词典等）或查阅任何书籍资料。
- 试题由洛谷网校学术组命制，欢迎报名洛谷网校第一轮课程。课程内容包含专题讲解、真题讲评与本试题讲评。https://class.luogu.com.cn/course/yugu25acs

一、单项选择题（共 15 题，每题 2 分，共计 30 分；每题有且仅有一个正确选项）

1．我国是人工智能大国，高度重视人工智能发展，积极推动互联网、大数据、人工智能和实体经济深度融合。下列关于人工智能领域的发展中，错误的说法是（ ）？

A．DeepSeek 和 Qwen 模型均为我国的开源大语言模型。

B．2025 世界人工智能大会（WAIC）于 7 月 26 日在上海开幕，展示了 AI 技术正突破屏幕界限，正在以各种形式深度融入物理世界。

C． 我国人工智能上市企业超过 300 家，且人工智能的创新领域分布广泛，包括但不限于自动驾驶、智能机器人等。

D．我国正大力推进算力设施的建设，但算力设施主要分布在东部地区，尚未对西部产业发展产生影响。

2．你使用 `g++ -O2 -o myapp main.cpp` 命令编译了一个程序，然后尝试用 `gdb ./myapp` 进行调试。在 GDB 中，当你试图使用 `print` 命令查看某个局部变量的值时，最有可能遇到的情况是？（ ）

A．程序立即崩溃，因为 GDB 无法处理优化过的代码。

B．变量值可以被正常显示，但执行速度会变慢。

C．GDB 会自动重新以 `-g` 选项编译该程序。

D．以上三者都错误。

3． 对于一个长度为 n 的，每一项取值在 $[1,k]$ 之间的正整数序列 a_i，下列三种排序算法在最坏情况下的时间复杂度，哪一个组合是**全部正确**的？（ ）

选项	快速排序	计数排序	堆排序
A	$\Theta(n\log n)$	$\Theta(k)$	$\Theta(n^2)$
B	$\Theta(n^2)$	$\Theta(n+k)$	$\Theta(n\log n)$

C	$\Theta(n\log n)$	$\Theta(n+k)$	$\Theta(n\log n)$
D	$\Theta(n^2)$	$\Theta(n\log n)$	$\Theta(n\log n)$

4. 使用 Kruskal 算法对一张连通图 $G=(V,E)$ 求最小生成树，且要求时间复杂度不超过 $O(|E|\log|E|)$。通常情况下，下列哪个数据结构是必要的？（ ）

A. 队列　　B. 链表　　C. 并查集　　D. 树状数组

5. 对于以下 C++ 程序，假设 n 为输入给定的正整数，不考虑整形变量溢出和编译器优化，则程序的时间复杂度是（ ）？

```
int i = 1;
while (i < n) {
    int j = n;
    while (j > 0)
        j /= 2;
    i = i * 2;
}
```

A. $\Theta(n)$　　B. $\Theta(n\log n)$　　C. $\Theta(\log^2 n)$　　D. $\Theta(\log n)$

6. 对于一张无自环的有向图 G，如果每对不同结点之间恰有一条有向边，则称 G 为竞赛图。对于一张有 5 个结点的竞赛图 G，下列哪一个数字可能是出度为 0 的结点的个数？（ ）

A. 4　　B. 3　　C. 2　　D. 1

7. 给定长度为 n 的正整数数组，使用前缀和与双指针算法，可以在线性时间解决哪一类问题？（ ）

A. 统计和为给定正整数 k 的子段数量　　B. 统计数组中逆序对个数

C. 统计第 k 小的元素及其个数　　D. 将数组从小到大排序

8. 给定一个后缀表达式 3 4 2 * 1 5 - 2 3 ^ ^ / +（其中：^ 表示乘方，例如 3^2=9），则这个后缀表达式的值下取整的结果是？（ ）

A. 2　　B. 3　　C. 4　　D. 65539

9. 假设 a,b,m 都是正整数，下列关于模运算的计算式中，恒成立的是（ ）？

A. $(a+b)\bmod m=(a\bmod m+b)\bmod m$

B. $(a-b)\bmod m=(a\bmod m-b)$

C. $(a/b)\bmod m=(a\bmod m)/(b\bmod m)$

D. $(a\times b)\bmod m\neq\big((a\bmod m)\times(b\bmod m)\big)\bmod m$

10．某递归函数的递推式为 $T(n)=3T(n/4)+\Theta(n^{1.2})$，则其渐进时间复杂度为（　）。（在本题中，记 $\log_4 3=0.79$）

A．$\Theta(n^{0.79})$　　B．$\Theta(n^{1.2})$　　C．$\Theta(n^{1.2}\log n)$　　D．$\Theta(n^{1.79})$

11．某校要从 5 名教师和 6 名学生中选派 5 人组成一个活动小组，要求：小组中至少包含 1 名教师和 2 名学生，且教师甲和学生乙不能同时入选。请问共有（　）种不同的选派方案？

A．345　　B．425　　C．341　　D．225

12．有一批正整数，均满足 $x\equiv 3\pmod 6$，现计划构造哈希函数 $h(x)=x \bmod m$，将元素放入 m 个桶内。若希望尽量覆盖全部的桶，以下哪一个 m 最合适？（　）

A．64　　B．72　　C．97　　D．120

13．在下列关于 Manacher 算法的描述中，对于一个长度为 N 的字符串 S，哪一个说法是正确的？（　）

A．该算法使用分治策略，并且平均时间复杂度为 $O(N\log N)$。

B．该算法通过动态规划实现，可在 $O(N^2)$ 时间求一个字符串的所有回文子串。

C．为了便于代码实现，通常需要在原字符串的每个字符之间和字符串首尾插入特殊字符。

D．为了降低时间复杂度，Manacher 算法需要预先使用 KMP 算法预处理 border 数组。这里，定义一个字符串 S 的 border 为 S 的一个**非 S 本身**的子串 T，满足 T 既是 S 的前缀，又是 S 的后缀。

14．对于有向图 $G=(V,E)$，定义 G_u 为将 G 忽略方向得到的无向图。现在对于图 G，有：

点集 $V=\{1,2,3,4,5,6\}$；

边集 $E=\{(1,2),(1,6),(3,2),(3,4),(5,4),(5,6)\}$，其中 (u,v) 表示 u 连向 v 的有向边。

则关于 G 和 G_u 的说法中，正确的是（　）？

A．G 是强连通图，而 G_u 是一张点双连通的二分图；

B．G 不是强连通图，而 G_u 是二分图，但是并不是点双连通的图；

C．G 不是强连通图，而 G_u 也不是二分图；

D．G 不是强连通图，而 G_u 是一张点双连通的二分图；

15．有 8 个布尔类型的变量 $x_1,x_2,\ldots,x_8$ 满足下列异或方程（$\oplus$ 代表异或运算）：

$$\begin{cases} x_1\oplus x_2\oplus x_3 \oplus x_5\oplus x_6\oplus x_7\oplus x_8=1\\ x_3\oplus x_6\oplus x_8=0\\ x_2\oplus x_3\oplus x_4\oplus x_5\oplus x_6=1\\ x_1\oplus x_2\oplus x_7=1\\ x_2\oplus x_6=0 \end{cases}$$

满足全部方程的**不同解的数量**为多少？（　）

A. 4　　B. 8　　C. 16　　D. 32

二、阅读程序（程序输入不超过数组或字符串定义的范围；判断题正确填 T，错误填 F；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

（1）

```
1 #include <iostream>
2 using namespace std;
3
4 const int maxn = 10000000 + 7;
5
6 int n, f[maxn], g[maxn];
7 long long m, ans = 1;
8
9 int main() {
10     cin >> n >> m;
11     for (int i = 1; i <= n; i++) {
12         cin >> f[i];
13     }
14     for (int i = 1; i <= n; i++) {
15         if(f[i] == 1)
16             for (int j = i; j <= n; j += i)
17                 g[j] = 1;
18         if(g[i] == 0)
19             ans = ans * i % m;
20     }
21     cout << ans << endl;
22     return 0;
23 }
```

假设输入的整数满足 $1 \le n \le 10^7, 1 \le m \le 10^9, f_i \in \{0, 1\}$，完成下面的判断题和单选题。

• 判断题

16. 该程序输出一行一个不小于 0 而小于 m 的整数。（　）

17. 删去程序的第 13 行和第 14 行，程序仍然能正常运行，且输出结果不变。（　）

18. 将第 15 行的 `if(f[i] == 1)` 改为 `if(f[i] == 1 && g[i] == 0)`，程序仍然能正常运行，且输出结果不变。（　）

• 单选题

19. 该程序（对于 n）的最坏情况下的时间复杂度是（　）？

A. $\Theta(n)$　　B. $\Theta(n \log\log n)$　　C. $\Theta(n \log n)$　　D. $\Theta(n^2)$

20. 对于以下输入，其输出的结果除以 13 的余数是（　）？

```
31 10000
0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

A. 0　　B. 1　　C. 2　　D. 3

（2）

```
#include <iostream>
#include <cstdio>
#include <cstring>
using namespace std;

const int N = 1010;

int n, m, mod;
long long ans, y[N], f[N], g[N], p[N], s[N];

long long qmi(long long a, int b, int p) {
    long long res = 1;
    while (b) {
        if (b & 1)
            res = res * a % p;
        a = a * a % p;
        b >>= 1;
    }
    return res;
}

long long solve(int n, int k) {
    memset(y, 0, sizeof(y));
    memset(f, 0, sizeof(f));
    memset(g, 0, sizeof(g));
    memset(p, 0, sizeof(p));
    memset(s, 0, sizeof(s));
```

```
28      long long ans = 0;
29      for (int i = 1; i <= k + 2; i++)
30          y[i] = (y[i - 1] + qmi(i, k, mod)) % mod;
31      f[0] = g[0] = 1;
32      for (int i = 1; i <= k + 2; i++)
33          f[i] = f[i - 1] * i % mod;
34      for (int i = 1; i <= k + 2; i++)
35          g[i] = -g[i - 1] * i % mod;
36      p[0] = s[k + 3] = 1;
37      for (int i = 1; i <= k + 2; i++)
38          p[i] = p[i - 1] * (n - i) % mod;
39      for (int i = k + 2; i >= 1; i--)
40          s[i] = s[i + 1] * (n - i) % mod;
41      for (int i = 1; i <= k + 2; i++) {
42          long long t1 = p[i - 1] * s[i + 1] % mod;
43          long long t2 = f[i - 1] * g[abs(k - i + 2)] % mod;
44          (ans += y[i] * t1 % mod * qmi(t2, mod - 2, mod) % mod) %= mod;
45          ans = (ans + mod) % mod;
46      }
47      return ans;
48  }
49
50  int main() {
51      scanf("%d%d%d", &n, &m, &mod);
52      long long res = 0;
53      for (int i = 1; i <= n; i++) {
54          res += (n - i + 1) * qmi(m, n - i, mod) % mod * solve(m, i);
55          res %= mod;
56      }
57      printf("%lld", res);
58      return 0;
59  }
```

假设输入的 n, m 是不超过 1000 的正整数，mod 是大于 10^4，但是不超过 10^9+7 的质数，完成下面的判断题和单选题。

• 判断题

21. qmi 函数实现了快速幂算法。该函数对于任意的不超过 10^9+7 的非负整数 b 和不超

过 10^9+7 的正整数 a,p 均能正确返回 a^b 对 p 取模的值。（　）

22．删除程序的第 23 到 27 行，程序仍然能正常运行，且输出结果不变。（　）

23．当输入的 m 为 1 时，输出总为 $\frac{n(n+1)}{2}$。（　）

• 单选题

24．这份程序的最坏时间复杂度是（　）。

A．$\Theta(n^2)$　　B．$\Theta(n\log n)$　　C．$\Theta(n^2\log n)$　　D．$\Theta(n^2\log(n\times \mathrm{mod}))$

25．对于以下的输入数据，输出结果为（　）。

```
3 4 10331
```

A．1900　　B．9500　　C．820　　D．1650

26．（4 分）这份程序求得了（ ）的答案对 mod 取模的结果。

A. 对于所有长度为 n 的，值域为 $[1,m]$ 的整数序列 a，计算 $\sum_{i=1}^{n}\sum_{j=i}^{n}\max_{k=i}^{j}a_k$ 之和。

B. 对于所有长度为 m 的，值域为 $[1,n]$ 的整数序列 a，计算 $\sum_{i=1}^{m}\sum_{j=i}^{m}\max_{k=i}^{j}a_k$ 之和。

C. 对于所有长度为 n 的，值域为 $[1,m]$ 的整数序列 a，计算 $\sum_{i=1}^{n}\sum_{j=i}^{n}\min_{k=i}^{j}a_k$ 之和。

D. 对于所有长度为 m 的，值域为 $[1,n]$ 的整数序列 a，计算 $\sum_{i=1}^{m}\sum_{j=i}^{m}\min_{k=i}^{j}a_k$ 之和。

（3）

```
 1 #include <iostream>
 2 #include <string>
 3
 4 typedef long long ll;
 5 const int alpha = 32;
 6 const int maxn = 10000 + 5, maxk = 100 + 5;
 7 const int maxv = 100000 + 5, maxs = 100000 + 5;
 8 const ll mod = 1000000000 + 7;
 9 using namespace std;
10
11 int n, k, v;
12 ll a[maxv];
13 string s;
14 int tr[maxs][alpha], dep[maxs], cnt[maxs];
15 int nxt[maxs], cnt1;
```

```
int id[maxs], cnt2;
ll c[maxn][maxk];
ll dp[maxn * 2][maxk], temp[maxk];
int siz[maxs];

void dfs(int ver) {
    int numb = 0, son = 0;
    for (int i = 0; i < alpha; i++) {
        if(tr[ver][i] != 0) {
            dfs(tr[ver][i]);
            ++numb;
            son = tr[ver][i];
        }
    }
    if (ver != 1 && numb == 1 && cnt[ver] == 0) {
        nxt[ver] = nxt[son];
    } else {
        nxt[ver] = ver;
        id[ver] = ++cnt2;
    }
}

void solve(int ver) {
    ll f = 1, g = 1;
    siz[ver] = cnt[ver];
    for (int i = 0; i <= siz[ver] && i <= k; i++) {
        dp[id[ver]][i] = f * c[siz[ver]][i] % mod;
        f = f * g % mod;
        g = g * a[dep[ver]] % mod;
    }
    for (int i = 0; i < alpha; i++)
        if(tr[ver][i] != 0)
            solve(nxt[tr[ver][i]]);
    for (int i = 0; i < alpha; i++)
        if (tr[ver][i] != 0) {
            int x = nxt[tr[ver][i]];
```

```
52            for (int j = 0; j <= k && j <= siz[ver] + siz[x]; j++)
53                temp[j] = 0;
54            ll value1 = 1;
55            for (int j1 = 0; j1 <= k && j1 <= siz[ver]; j1++) {
56                ll value2 = dp[id[ver]][j1];
57                for (int j2 = 0; j2 <= siz[x] && j1 + j2 <= k; j2++) {
58                    temp[j1 + j2] = (temp[j1 + j2] +
59                                     value2 * dp[id[x]][j2]) % mod;
60                    value2 = value2 * value1 % mod;
61                }
62                value1 = value1 * a[dep[ver]] % mod;
63            }
64            siz[ver] += siz[x];
65            for (int j = 0; j <= siz[ver] && j <= k; j++)
66                dp[id[ver]][j] = temp[j];
67        }
68 }
69
70 int main() {
71     cin >> n >> k >> v;
72     for (int i = 0; i <= v; i++)
73         cin >> a[i];
74     cnt1 = 1;
75     for (int i = 1; i <= n; i++) {
76         cin >> s;
77         int cur = 1, len = s.size();
78         for (int j = 0; j < len; j++) {
79             int ch = s[j] & 31;
80             if (tr[cur][ch] == 0) {
81                 tr[cur][ch] = ++cnt1;
82                 dep[cnt1] = dep[cur] + 1;
83             }
84             cur = tr[cur][ch];
85         }
86         ++cnt[cur];
87     }
```

```
88      c[0][0] = 1;
89      for (int i = 1; i <= n; i++) {
90          c[i][0] = 1;
91          for (int j = 1; j <= k && j <= i; j++)
92              c[i][j] = (c[i-1][j-1] + c[i-1][j]) % mod;
93      }
94      dfs(1);
95      solve(1);
96      cout << dp[id[1]][k] << endl;
97      return 0;
98  }
```

设 $1 \le n \le 10^4$，$1 \le k \le \min(n, 10^2)$，$1 \le v \le 10^5$，$0 \le a_i < 10^9 + 7$，所有 s 均为长度在 1 到 v 之间的小写英文字母组成的字符串，记所有 s 的长度之和为 L，假设 $L \le 10^5$，完成下面的判断题和单选题。

• 判断题

27. 若输入的所有字符串长度均不小于 2，且它们的第一个字符均相同，第二个字符两两不同，且 $a_1 = 1$，则输出的结果为 $\frac{n!}{k!(n-k)!}$。（ ）

28. 将第 5 行的 `const int alpha = 32;` 改为 `const int alpha = 26;` 后，程序仍然能正常运行，且输出结果不变。（ ）

29. （2 分）若将第 18 行的 `ll dp[maxn * 2][maxk]` 改为 `ll dp[maxs][maxk]`，第 30 行的 `if (ver != 1 && numb == 1 && cnt[ver] == 0)` 改为 `if (false)`，程序仍然能正常运行，且输出结果不变。（ ）

• 选择题

30. 假设 alpha 为常数，那么该程序的最坏时间复杂度是（ ）。

A. $\Theta(v + Lk + nk^2)$　B. $\Theta(v + L + nk^2)$　C. $\Theta(v + Lk)$　D. $\Theta(v + L + nk)$

31. （4 分）记 c 为满足在程序运行结束时满足恰有 26 个 $j \in \{0,1,\cdots,31\}$ 满足 $tr_{i,j} \neq 0$ 的 i 的个数。若在本题条件下，记 c 的最大值为 c_1，而在满足本题条件且 $L \le 10^4$ 时，记此时 c 的最大值为 c_2。设 $c_0 = c_1 - c_2$，则 c_0 除以 17 的余数是（ ）。

A. 0　B. 4　C. 7　D. 11

32. 对于以下的输入数据，程序输出的结果是（ ）。

```
10 3 12
3 1 4 6 5 2 0 3 6 1 4 2 5
```

```
ak
firstround
goodluck
havefun
luogu
luogudevteam
luotianyi
rpplusplus
scp
scps
```

A. 3248　　B. 3617　　C. 3816　　D. 4068

三、完善程序（单选题，每小题 3 分，共计 30 分）

（1）（**树的重心**）给定一颗树，树中包含 n 个结点（编号 $1 \sim n$）和 $n-1$ 条无向边。请你找到树的**重心**，并输出将重心删除后，剩余各个连通块中点数的最大值。**重心**是指树中的一个结点，如果将这个点删除后，剩余各个连通块中点数的**最大值最小**，那么这个节点被称为树的重心。上述参数满足 $1 \leq n \leq 2 \times 10^5$。试补全程序。

```
1 #include <iostream>
2 #include <algorithm>
3 using namespace std;
4
5 const int N = 2e5 + 10;
6
7 int n, pos, maxx_part = 0x3f3f3f3f, siz[N], v[N];
8 int h[N], e[N], ne[N], idx;
9
10 void add(int a, int b) {
11     e[idx] = b;
12     ne[idx] = h[a];
13     ① ;
14 }
15
16 void dfs(int x) {
17     v[x] = 1;
18     ②;
```

```
19      int max_part = 0;
20      for (int i = h[x]; ~i; i = ne[i]) {
21          int j = e[i];
22          if (v[j]) continue;
23          dfs(j);
24          siz[x] += siz[j];
25          max_part = max(max_part, siz[j]);
26      }
27      ③;
28      maxx_part = min(maxx_part, max_part);
29  }
30
31  int main() {
32      ④;
33      cin >> n;
34      for (int i = 0; i < n; i++) {
35          int a, b;
36          scanf("%d%d", &a, &b);
37          add(a, b), add(b, a);
38      }
39      dfs(1);
40      cout << ⑤;
41      return 0;
42  }
```

33. ①处应填（　）

A. h[a] = ne[idx + 1]　　B. h[a] = e[idx + 1]

C. h[a] = ++idx　　D. h[a] = idx++

34. ②处应填（　）

A. siz[x] = -1　　B. siz[x] = 0

C. siz[x] = 1　　D. siz[x] = 0x3f3f3f3f

35. ③处应填（　）

A. max_part = max(max_part, n - siz[x])

B. maxx_part = max(maxx_part, n - siz[x])

C. max_part = max(max_part, siz[x])

D. maxx_part = max(maxx_part, siz[x])

36. ④处应填（　）。

A. memset(h, 0, sizeof(h))　　B. memset(h, 63, sizeof(h))

C. memset(h, -63, sizeof(h))　　D. memset(h, 255, sizeof(h))

37. ⑤处应填（　）。

A. siz[maxx_part == 2e9 ? 0 : maxx_part]

B. maxx_part

C. maxx_part + 1

D. 2 * maxx_part - 1

（2）（**矩形覆盖问题**）在平面直角坐标系上有 n 个矩形，矩形的每条边与坐标系的横轴平行或垂直。对于第 i 个矩形，矩形左下角的顶点坐标为 (s_x, s_y)，右上角的顶点坐标为 (e_x, e_y)。每个矩形要么严格包含另一个矩形，要么完全不相交。矩形的边界不会相交、顶点或边也不会重合。要求，对于每个矩形，分别求出包含它的矩形中最小的矩形。如果没有被覆盖，则输出 0。上述参数满足 $1 \le n \le 2 \times 10^5$，$1 \le s_x, s_y, e_x, e_y \le 10^9$，你需要设计一个时间复杂度不超过 $\Theta(n \log n)$ 的程序。试补全程序。

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <utility>
using namespace std;

typedef long long ll;
typedef pair<ll, ll> pii;

struct rectangle {
    ll sx, sy, ex, ey;
};

struct line {
    ll x, sy, ey, id;
    friend bool operator < (const line &lhs, const line &rhs) {
        return ①;
    }
};

struct segmentTree {
    vector<int> tree, lazy;

```

```
24    static int ls(int u) { return u << 1; }
25    static int rs(int u) { return u << 1 | 1; }
26    segmentTree(int N): tree(N * 4 + 1, 0), lazy(N * 4 + 1, -1) {}
27
28    void pushdown(int p) {
29       if(~lazy[p]) {
30          tree[ls(p)] = tree[rs(p)] = lazy[p];
31          lazy[ls(p)] = lazy[rs(p)] = lazy[p];
32          lazy[p] = -1;
33       }
34    }
35
36    void update(int p, int l, int r, int ql, int qr, int k) {
37       if (l >= ql && r <= qr) {
38          tree[p] = lazy[p] = k;
39          return;
40       }
41       pushdown(p); int mid = (l + r) >> 1;
42       if (②) update(ls(p), l, mid, ql, qr, k);
43       if (qr > mid)  update(rs(p), mid + 1, r, ql, qr, k);
44    }
45
46    int query(int p, int l, int r, int x) {
47       if (l == r) return tree[p];
48       pushdown(p); int mid = (l + r) >> 1;
49       if (mid >= x) return query(ls(p), l, mid, x);
50       else return query(rs(p), mid + 1, r, x);
51    }
52 };
53
54 template<class T> struct discretize {
55    vector<T> data;
56    discretize(): data() {}
57
58    void insert(const T &x) { data.push_back(x); }
```

```
59
60      void build() {
61          sort(data.begin(), data.end());
62          auto endp = unique(data.begin(), data.end());
63          data.erase(endp, data.end());
64      }
65
66      size_t size() const { return data.size(); }
67
68      int operator[](const T &x) {
69          return ③;
70      }
71  };
72
73  int main() {
74      int n; cin >> n;
75      vector<rectangle> P(n);
76      for (int i = 0; i < n; i++)
77          cin >> P[i].sx >> P[i].sy >> P[i].ex >> P[i].ey;
78      discretize<ll> discx, discy;
79      for (int i = 0; i < n; i++) {
80          discx.insert(P[i].sx);
81          discx.insert(P[i].ex);
82          discy.insert(P[i].sy);
83          discy.insert(P[i].ey);
84      }
85      discx.build(); discy.build();
86      for (int i = 0; i < n; i++) {
87          P[i].sx = discx[P[i].sx] + 1;
88          P[i].ex = discx[P[i].ex] + 1;
89          P[i].sy = discy[P[i].sy] + 1;
90          P[i].ey = discy[P[i].ey] + 1;
91      }
92      vector<line> L; int cnt = 0;
93      for(int i = 0; i < n; i++) {
```

```
94        L.push_back({P[i].sx, P[i].sy, P[i].ey, ++cnt});
95        L.push_back({④});
96    }
97    sort(L.begin(), L.end());
98    int N = discx.size();
99    segmentTree S(N);
100   vector<int> fa(n + 1, -1);
101   for (int i = 0; i < L.size(); i++) {
102       int c = S.query(1, 1, N, L[i].sy);
103       if (L[i].id < 0)
104           S.update(1, 1, N, L[i].sy, L[i].ey, fa[-L[i].id]);
105       else {
106           fa[L[i].id] = c;
107           ⑤
108       }
109   }
110   for (int i = 1; i <= cnt; ++i)
111       cout << fa[i] << " \n"[i == cnt];
112   return 0;
113 }
```

38. ①处应填（　）

A. lhs.sy < rhs.sy || lhs.ey < rhs.ey

B. lhs.sy == rhs.sy? lhs.id < rhs.id : lhs.sy < rhs.sy

C. lhs.ey == rhs.ey ? lhs.id < rhs.id : lhs.ey < rhs.ey

D. lhs.x == rhs.x ? lhs.id < rhs.id : lhs.x < rhs.x

39. ②处应填（　）

A. ql < mid　　B. ql <= mid　　C. ql >= mid　　D. ql == mid

40. ③处应填（　）

A. lower_bound(data.begin(), data.end(), x) - data.begin()

B. upper_bound(data.begin(), data.end(), x) - data.begin()

C. find(data.begin(), data.end(), x) - data.begin()

D. distance(data.end(), upper_bound(data.begin(), data.end(), x))

41. ④处应填（　）

A. P[i].sx, P[i].sy, P[i].ey, cnt

B. P[i].sx, P[i].sy, P[i].ey, -cnt

C. P[i].ex, P[i].sy, P[i].ey, cnt

D. P[i].ex, P[i].sy, P[i].ey, -cnt

42. ⑤处应填（　）

```
A. fa[i] = S.query(1, 1, N, c)
B. fa[i] = S.query(1, 1, N, L[i].id)
C. S.update(1, 1, N, L[i].sy, L[i].ey, L[i].id)
D. S.update(1, 1, N, L[i].sx, L[i].ex, L[i].id)
```

2025 LUOGU 非专业级别收容能力认证第一轮

（SCP-S1）提高级 C++语言试题答案

试题由洛谷网校学术组命制，欢迎报名洛谷网校第一轮课程。课程内容包含专题讲解、真题讲评与本试题讲评。https://class.luogu.com.cn/course/yugu25acs

一、单项选择题（共 15 题，每题 2 分，共计 30 分；每题有且仅有一个正确选项）

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

二、阅读程序（判断题正确填 T，错误填 F；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

第一题	判断题			选择题	
	1	2	3	4	5
	F	T	T	C	B

第二题	判断题			选择题		
	1	2	3	4	5	6（4 分）
	F	T	F	D	C	C

第三题	判断题			选择题		
	1	2	3（2 分）	4	5（4 分）	6
	T	F	T	D	B	C

三、完善程序（单选题，每小题 3 分，共计 30 分）

第一题					第二题				
1	2	3	4	5	1	2	3	4	5
D	C	A	D	B	D	B	A	D	C

欢迎关注洛谷科技公众号。