

CSP-S 2026

第一轮真题 · 试卷与答案

作答时间: 120 分钟 姓名: _____

满分 100 分, 共 43 个作答点。判断题填写 A 或 B; 参考答案附于卷末。

一、单项选择 (共 30 分)

第 1 题 (2 分)

执行下列代码后, cnt 的值是 ()。

```
1 int x = 2026, cnt = 0;
2 while (x) {
3     x &= x - 1;
4     cnt++;
5 }
```

- | | |
|--------------|-------------|
| A. 6 | B. 7 |
| C. 11 | D. 8 |
-

第 2 题 (2 分)

用权值 {1, 2, 3, 4, 5, 6, 7, 8} 构造哈夫曼树, 其带权路径长度是 ()。

- | | |
|---------------|---------------|
| A. 108 | B. 96 |
| C. 99 | D. 102 |
-

第 3 题 (2 分)

把 1 到 1000 的所有整数按十进制写出, 数字 “1” 总共出现了多少次 ()。

- | | |
|---------------|---------------|
| A. 300 | B. 271 |
| C. 301 | D. 320 |
-

第 4 题 (2 分)

将 5 封信随机装入 5 个写好地址的信封 (每封一个), 恰好有 2 封装对的方案数是 ()。

- | | |
|--------------|--------------|
| A. 44 | B. 24 |
| C. 10 | D. 20 |
-

第 5 题 (2 分)

$3^{2026} \bmod 100$ 的值是 ()。

- | | |
|-------|-------|
| A. 29 | B. 9 |
| C. 43 | D. 81 |
-

第 6 题 (2 分)

有 5 堆石子排成一行，重量依次为 4、1、3、2、5。每次只能把相邻的两堆合并成一堆，代价为这两堆重量之和。将所有石子合并成一堆的最小总代价是 ()。

- | | |
|-------|-------|
| A. 36 | B. 35 |
| C. 34 | D. 33 |
-

第 7 题 (2 分)

树状数组维护长度 $n = 16$ 的序列，查询前缀和 $\text{sum}(11)$ 与单点修改 $\text{add}(3, x)$ 分别需要访问树状数组中多少个下标 ()。

- | | |
|----------|----------|
| A. 3 和 4 | B. 4 和 4 |
| C. 3 和 5 | D. 4 和 3 |
-

第 8 题 (2 分)

有向无环图 G 顶点集为 $1, 2, 3, 4$ ，边集为 $(1, 2), (1, 3)$ ，顶点 4 与任何顶点均不相邻。该图不同的拓扑序共有多少种 ()。

- | | |
|-------|------|
| A. 12 | B. 8 |
| C. 4 | D. 6 |
-

第 9 题 (2 分)

某分治算法满足 $T(n) = T(n/3) + T(2n/3) + \Theta(n)$ ， $T(1) = O(1)$ ，则 $T(n)$ 是 ()。

- | | |
|-----------------------|------------------|
| A. $\Theta(n \log n)$ | B. $\Theta(n^2)$ |
| C. $\Theta(n^{1.5})$ | D. $\Theta(n)$ |
-

第 10 题 (2 分)

无根树含 9 个结点（编号为 1—9），边集为 $(1, 2), (1, 3), (2, 4), (2, 5), (3, 6), (6, 7), (7, 8), (5, 9)$ 。该树的直径（以边数计）与重心分别是 ()。

- | | |
|-----------------|-----------------|
| A. 直径 6，重心为结点 3 | B. 直径 7，重心为结点 2 |
| C. 直径 8，重心为结点 1 | D. 直径 7，重心为结点 1 |
-

第 11 题 (2 分)

一张有向图缩点后得到的有向无环图含 6 个顶点，其中入度为 0 的顶点有 3 个，出度为 0 的顶点有 4 个。为使原图变成强连通图，至少需要添加多少条有向边 ()。

- | | |
|------|------|
| A. 7 | B. 6 |
| C. 4 | D. 3 |
-

第 12 题 (2 分)

含 6 个结点的不同形态的二叉树共有多少棵 (结点不带标号，区分左右子树) ()。

- | | |
|--------|--------|
| A. 42 | B. 429 |
| C. 132 | D. 720 |
-

第 13 题 (2 分)

字符串 $S = \text{"ababaabab"}$ ，其所有既是真前缀又是真后缀的子串 (非空) 的长度之和是 ()。

- | | |
|------|------|
| A. 4 | B. 6 |
| C. 7 | D. 5 |
-

第 14 题 (2 分)

用归并排序统计逆序对，合并部分的核心代码为：

若把判断条件中的 $a[i] \leq a[j]$ 改成 $a[i] < a[j]$ ，则 ans 统计出的结果 ()。

```
1 // 分并 a[l..mid] 与 a[mid+1..r]，同时累加逆序对
2 if (a[i] <= a[j]) {
3     tmp[k++] = a[i++]; // 取左半段元素
4 } else {
5     tmp[k++] = a[j++]; // 取右半段元素
6     ans += mid - i + 1;
7 }
```

- A. 完全不变
- B. 变为原来的两倍
- C. 变为满足 $i < j$ 且 $a[i] \geq a[j]$ 的数对个数
- D. 变为原来的一半
-

第 15 题 (2 分)

执行 `power(2, 100, 1000)` 调用下列函数, 返回值是 ()。

```
1 long long power(long long a, long long b, long long p) {
2     long long r = 1 % p;
3     while (b) {
4         if (b & 1)
5             r = r * a % p;
6         a = a * a % p;
7         b >>= 1;
8     }
9     return r;
10 }
```

A. 576

B. 376

C. 976

D. 176

二、阅读程序 (共 40 分)

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

第 16 题—第 21 题

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 int a[100];
5 string s;
6 int gen[13] = {1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1};
7 int main() {
8     cin >> s;
9     for (int i = 0; i < 32; ++i) {
10         a[i] = s[i] - '0';
11     }
12     for (int i = 32; i < 44; ++i) {
13         a[i] = 0;
14     }
15     for (int i = 0; i < 32; ++i) {
16         if (a[i] == 0) continue;
17         for (int j = 0; j < 13; ++j) {
18             a[i + j] ^= gen[j];
19         }
20     }
21     for (int i = 32; i < 44; ++i) {
22         cout << a[i];
23     }
24     cout << endl;
25     return 0;
26 }
```

说明: 输入保证为一个长度恰为 32 的 '0' / '1' 字符串。

第 16 题 (1 分) (1 分) 当输入为 32 个 '0' 时, 程序输出 12 个 0。 ()

- A. 正确 B. 错误
-

第 17 题 (1.5 分) 程序运行结束后, 数组 a 中下标从 0 到 31 的元素一定全部为 0。 ()

- A. 正确 B. 错误
-

第 18 题 (1.5 分) 若将第 12—14 行 (为 a[32] 到 a[43] 补 0 的循环) 删除, 会改变程序输出的结果。 ()

- A. 正确 B. 错误
-

第 19 题 (3 分) 关于第 6 行定义的数组 gen, 下列说法正确的是 ()。

- A. gen 共有 12 个元素, 表示一个 12 位的除数
B. gen 共有 13 个元素, 表示一个 13 位的被除数
C. gen 共有 13 个元素, 其中 gen[0] 是除数的最高位
D. gen 共有 13 个元素, 其中 gen[12] 是除数的最高位
-

第 20 题 (3 分) 该程序实现的功能, 最准确的说法是 ()。

- A. 将输入的 32 位串看成二进制数 M , 输出 M 与 13 位二进制数 1100000001111 按位异或的结果
B. 将输入串视为 32 位二进制数 M , 在其后补 12 个 0 (即计算 $M \times 2^{12}$), 再对它做模 2 除法求余数, 并输出 12 位余数
C. 对输入的 32 位串逐位取反并输出结果
D. 统计输入串中 1 的个数, 并把该个数用 12 位二进制表示后输出
-

第 21 题 (3 分) 若把第 16 行 if (a[i] == 0) continue; 删除, 说法正确的是 ()。

- A. 程序输出的结果不会改变
B. 可能造成程序运行错误
C. 程序能够正常输出一个 12 位 '0' / '1' 串, 但是输出结果与输入的 s 无关
D. 程序运行结束后, a[0] 的值一定为 0
-

第 22 题—第 27 题

```
1  #include <iostream>
2  using namespace std;
3  int n, m, a[100007], L, R, lg[100007], i, j, t, dp[100007][25], pw[25];
4  int gcd(int x, int y) {
5      if (y == 0) return x;
6      return gcd(y, x % y);
7  }
8  int main() {
9      cin >> n >> m;
10     for (i = 1; i <= n; i++) cin >> a[i];
11     t = 0;
12     pw[0] = 1;
13     for (i = 1; i <= 24; i++) pw[i] = pw[i - 1] * 2;
14     for (i = 1; i <= 100000; i++)
15         if (pw[t + 1] >= i) lg[i] = t;
16         else { t++; lg[i] = t; }
17     for (i = 1; i <= n; i++) {
18         dp[i][0] = a[i];
19     }
20     for (j = 1; j <= lg[n]; j++) {
21         for (i = 1; i + pw[j] - 1 <= n; i++) {
22             dp[i][j] = gcd(dp[i][j - 1], dp[i + pw[j] - 1][j - 1]);
23         }
24     }
25     for (i = 1; i <= m; i++) {
26         cin >> L >> R;
27         cout << gcd(dp[L][lg[R - L + 1]], dp[R - pw[lg[R - L + 1]] + 1][lg[R - L + 1]]) << endl;
28     }
29     return 0;
30 }
```

说明：保证 $1 \leq n \leq 100000$ ，每次查询满足 $1 \leq L \leq R \leq n$ ，且数组 a 的元素均为正整数。

第 22 题 (1.5 分) 当 $n = 5$ ， $a = 4, 2, 6, 3, 3$ ，且仅有一次查询 $L = 2, R = 5$ 时，输出为 1。 ()

- A. 正确 B. 错误

第 23 题 (1.5 分) 当某次查询的区间长度为 1 (即 $L = R$) 时，该次查询的输出一定等于 $a[L]$ 。 ()

- A. 正确 B. 错误

第 24 题 (1.5 分) 任意一次查询的输出结果一定不小于该查询区间内的最小值。 ()

- A. 正确 B. 错误

第 25 题 (3 分) 对于 $j \geq 1$ ，数组 $dp[i][j]$ 保存的是 ()。

- A. 从 $a[i]$ 开始连续 j 个数的最大公约数
B. 从 $a[i]$ 开始连续 2^j 个数的最大公约数
C. $a[i]$ 与 $a[j]$ 的最大公约数
D. 从 $a[1]$ 到 $a[i]$ 的最大公约数

第 26 题 (3 分) 若把一次求最大公约数的运算视为 $O(1)$ ，则第 17—22 行建表过程的时间复杂度为 ()。

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

第 27 题 (3 分) 设 x 为一次查询的区间长度 (即 $x = R - L + 1$)，则使得 $\lg[x] = 5$ 的 x 的取值范围是 ()。

- A. $[16, 31]$ B. $[17, 32]$
C. $[32, 63]$ D. $[33, 64]$

第 28 题—第 33 题

```
1  #include <iostream>
2  using namespace std;
3  int n, fa[100007], f[100007], ans;
4  int main() {
5      cin >> n;
6      for (int i = 2; i <= n; ++i) {
7          cin >> fa[i];
8      }
9      for (int i = n; i >= 2; --i) {
10         if (f[fa[i]] + f[i] + 1 > ans) {
11             ans = f[fa[i]] + f[i] + 1;
12         }
13         if (f[i] + 1 > f[fa[i]]) {
14             f[fa[i]] = f[i] + 1;
15         }
16     }
17     cout << ans << endl;
18     return 0;
19 }
```

说明：输入第一行为结点个数 n ，第二行为 $n - 1$ 个整数，依次表示结点 $2 \sim n$ 的父结点编号，满足 $2 \leq n \leq 10000$ 且 $1 \leq fa[i] < i$ ，根结点为 1。

第 28 题 (1.5 分) 当 $n = 5$ ， $fa[2] \sim fa[5] = 1, 2, 3, 4$ 时，程序输出 4。 ()

- A. 正确 B. 错误

第 29 题 (1.5 分) 程序输出前， $f[1]$ 的值一定等于 ans 的值。 ()

- A. 正确 B. 错误

第 30 题 (1.5 分) 将第 10—12 行与第 13—15 行两个 if 语句的顺序交换后，程序输出结果不受影响。 ()

- A. 正确 B. 错误

第 31 题 (3 分) 程序输出的 ans 表示的是 ()。

- A. 树中距离最远的两个结点之间路径所经过的边数 B. 根结点 1 到最远叶子结点之间路径所经过的边数
- C. 树中叶子结点的个数 D. 所有结点的父结点编号之和
-

第 32 题 (3 分) 当 $n = 7$, $fa[2] \sim fa[7] = 1, 1, 2, 2, 3, 3$ 时, 输出为 ()。

- A. 2 B. 3
- C. 4 D. 5
-

第 33 题 (3 分) 当 $n = 10$, 满足输出为 9 的合法输入种类数为 ()。

- A. 0 B. 9
- C. 256 D. 512
-

三、完善程序 (共 30 分)

(单选题, 每小题 3 分, 共计 30 分)

第 34 题—第 38 题

给定一张有 n 个顶点、 m 条边的无向图, 每条边带有符号 '+' 或 '-'。对于一条从顶点 s 到顶点 t 的路线, 允许重复经过顶点和边。定义一条路线的权值如下: 记 n^+, n^- 分别为经过的 '+' 边数和经过的 '-' 边数, 则该路线的权值为 $|n^+ - n^-|$ 。

请计算从 s 到 t 的路线的最小权值。若不存在从 s 到 t 的路线, 则输出 -1。

输入第一行为四个整数 n, m, s, t 。接下来 m 行, 每行给出两个整数 a, b 和一个字符 '+' 或 '-', 描述一条连接 a 与 b 的无向边及其符号。

数据满足 $2 \leq n \leq 2 \times 10^5$, $1 \leq m \leq 4 \times 10^5$, $1 \leq s, t \leq n$ 且 $s \neq t$, $1 \leq a, b \leq n$, 可能出现重边。

以下程序通过 BFS 求出最小权值。请补全程序。

```
1  #include <iostream>
2  constexpr int N = 200005;
3  constexpr int M = 400005;
4  int n, m, s, t;
5  int h[N], e[M << 1], ne[M << 1], w[M << 1], idx;
6  int q[N], d[N], c[N];
7  void add(int a, int b, int z) {
8      e[idx] = b;
9      w[idx] = z;
10     ne[idx] = h[a];
11     h[a] = idx++;
12 }
13 int main() {
14     std::cin >> n >> m >> s >> t;
15     for (int i = 1; i <= n; i++)
16         h[i] = d[i] = c[i] = -1;
17     for (int i = 0; i < m; i++) {
18         int a, b;
19         char op[2];
20         std::cin >> a >> b >> op;
21         int z = /* ① */;
22         add(a, b, z);
23         add(b, a, z);
24     }
25     int hh = 0, tt = 0;
26     int p = 0, ng = 0, ok = 1;
27     q[tt++] = s;
28     d[s] = c[s] = 0;
29     while (/* ② */) {
30         int x = q[hh++];
31         for (int i = h[x]; i != -1; i = ne[i]) {
32             int y = e[i];
33             if (w[i] > 0) p = 1;
34             if (w[i] < 0) ng = 1;
35             if (d[y] == -1) {
36                 d[y] = /* ③ */;
37                 c[y] = c[x] ^ 1;
38                 q[tt++] = y;
```

```

39         } else if (/* ④ */)
40             ok = 0;
41     }
42 }
43 if (d[t] == -1) {
44     std::cout << -1;
45     return 0;
46 }
47 if (!p || !ng) {
48     std::cout << d[t];
49     return 0;
50 }
51 if (/* ⑤ */) std::cout << 0;
52 else std::cout << 1;
53 return 0;
54 }

```

第 34 题 (3 分) ①处应填 ()。

- | | |
|---------------------------------------|--------------------------------------|
| A. <code>op[0] == '+' ? 0 : 1</code> | B. <code>op[0] == '+'</code> |
| C. <code>op[0] == '+' ? 1 : -1</code> | D. <code>op[0] == '-' ? 1 : 0</code> |

第 35 题 (3 分) ②处应填 ()。

- | | |
|-----------------------------|----------------------------|
| A. <code>hh < n</code> | B. <code>tt < n</code> |
| C. <code>hh <= tt</code> | D. <code>hh < tt</code> |

第 36 题 (3 分) ③处应填 ()。

- | | |
|--------------------------|--------------------------|
| A. <code>d[y] + 1</code> | B. <code>d[x] + 1</code> |
| C. <code>d[x]</code> | D. <code>d[x] - 1</code> |

第 37 题 (3 分) ④处应填 ()。

- | | |
|------------------------------|----------------------------------|
| A. <code>c[y] == c[x]</code> | B. <code>w[i] == 1</code> |
| C. <code>c[y] != c[x]</code> | D. <code>d[y] + 1 != d[x]</code> |

第 38 题 (3 分) ⑤处应填 ()。

- | | |
|--|---|
| A. <code>ok && c[s] == c[t]</code> | B. <code>ok && c[s] != c[t]</code> |
| C. <code>!ok c[s] == c[t]</code> | D. <code>!ok && c[s] != c[t]</code> |

第 39 题—第 43 题

给定 n 名学生参加一场考试，考试共有 m 道选择题，每道题只有 A、B 两个选项。

第 i 名学生的作答为一个长度为 m 的字符串。若最终公布的标准答案与该学生在某道题上的作答相同，则该学生在这道题上得 1 分，否则不得分。记第 i 名学生最终得到的总分为 r_i 。

每名学生还有一个预期得分 x_i 。现在需要构造一份标准答案，使尽可能大。

数据满足 $1 \leq n \leq 18, 1 \leq m \leq 300, 0 \leq x_i \leq m$ 。

提示：可以换一个角度处理 $\sum_{i=1}^n |r_i - x_i|$ ，把它写成更易优化的形式；对正整数 x ，`__builtin_ctzll(x)` 返回 x 的二进制表示末尾连续 0 的个数；`__builtin_popcountll(x)` 返回 x 的二进制表示中 1 的个数。

以下程序构造出一组满足要求的标准答案。请补全程序。

```
1  #include <cstdlib>
2  #include <iostream>
3  #include <string>
4  #include <vector>
5  using namespace std;
6  typedef long long ll;
7  typedef unsigned long long ull;
8  int main() {
9      int n, m;
10     cin >> n >> m;
11     vector<ll> x(n), c(n);
12     for (int i = 0; i < n; i++) {
13         cin >> x[i];
14         c[i] = /* ① */;
15     }
16     vector<string> a(n);
17     for (int i = 0; i < n; i++)
18         cin >> a[i];
19     vector<int> s(n, -1);
20     vector<ll> q(m, 0);
21     ll C = 0, S = 0;
22     for (int i = 0; i < n; i++) {
23         C -= c[i];
24         for (int j = 0; j < m; j++) {
25             if (a[i][j] == 'A') q[j]--;
26             else q[j]++;
27         }
28     }
29     for (int j = 0; j < m; j++) S += abs(q[j]);
30     ll ans = C + S;
31     ull best = 0, lst = 0;
32     for (ull mask = 1; mask < (1ULL << n); mask++) {
33         ull g = /* ② */;
34         ull d = g ^ lst;
35         int k = /* ③ */;
36         C -= /* ④ */;
37         for (int j = 0; j < m; j++) {
38             ll old = q[j];
39             int v = (a[k][j] == 'A' ? 1 : -1);
40             q[j] -= 2ll * s[k] * v;
```

```

41         S += abs(q[j]) - abs(old);
42     }
43     s[k] = -s[k];
44     if (C + S > ans) {
45         ans = C + S;
46         best = g;
47     }
48     lst = g;
49 }
50 for (int i = 0; i < n; i++) {
51     if ((best >> i) & 1) s[i] = 1;
52     else s[i] = -1;
53 }
54 string res(m, 'A');
55 for (int j = 0; j < m; j++) {
56     ll v = 0;
57     for (int i = 0; i < n; i++) {
58         if (a[i][j] == 'A') v += s[i];
59         else v -= s[i];
60     }
61     if (/* ⑤ */) res[j] = 'A';
62     else res[j] = 'B';
63 }
64 cout << res << endl;
65 return 0;
66 }

```

第 39 题 (3 分) ①处应填 ()。

- | | |
|-------------------|------------------------|
| A. $2 * x[i] - m$ | B. $-m + 2 * x[i] + 1$ |
| C. $m - 2 * x[i]$ | D. $m + 2 * x[i]$ |

第 40 题 (3 分) ②处应填 ()。

- | | |
|--|---|
| A. $\text{mask} (\text{mask} >> 1)$ | B. $\text{mask} ^ (\text{mask} >> 1)$ |
| C. $\text{mask} \& (\text{mask} >> 1)$ | D. $\text{mask} ^ ((\text{mask} >> 1) + 1)$ |

第 41 题 (3 分) ③处应填 ()。

- | | |
|-------------------------------------|--------------------------------------|
| A. $_\text{builtin_ctzll}(d) + 1$ | B. $_\text{builtin_popcountll}(d)$ |
| C. $_\text{builtin_ctzll}(g)$ | D. $_\text{builtin_ctzll}(d)$ |

第 42 题 (3 分) ④处应填 ()。

- | | |
|--------------------------|------------------|
| A. $2ll * s[k] * c[k]$ | B. $s[k] * c[k]$ |
| C. $2ll * (s[k] - c[k])$ | D. $2ll * c[k]$ |

第 43 题 (3 分) ⑤处应填 () 。

A. $v \geq (n \& 1)$

B. $v > (n \& 1)$

C. $v + (n \& 1) \geq 0$

D. $v * (n \& 1) \geq 0$

参考答案

满分 100 分。判断题 A 表示正确，B 表示错误。

题号	分值	答案	题号	分值	答案
第 1 题	2	D	第 2 题	2	D
第 3 题	2	C	第 4 题	2	D
第 5 题	2	A	第 6 题	2	C
第 7 题	2	A	第 8 题	2	B
第 9 题	2	A	第 10 题	2	D
第 11 题	2	C	第 12 题	2	C
第 13 题	2	B	第 14 题	2	C
第 15 题	2	B	第 16 题	1	A
第 17 题	1.5	A	第 18 题	1.5	B
第 19 题	3	C	第 20 题	3	B
第 21 题	3	C	第 22 题	1.5	A
第 23 题	1.5	A	第 24 题	1.5	B
第 25 题	3	B	第 26 题	3	B
第 27 题	3	D	第 28 题	1.5	A
第 29 题	1.5	B	第 30 题	1.5	B
第 31 题	3	A	第 32 题	3	C
第 33 题	3	C	第 34 题	3	C
第 35 题	3	D	第 36 题	3	B
第 37 题	3	A	第 38 题	3	C
第 39 题	3	C	第 40 题	3	B
第 41 题	3	D	第 42 题	3	A
第 43 题	3	A			