

CSP-J 模拟 (五)

2025年8月2日

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

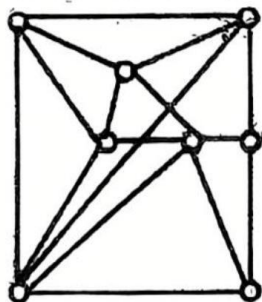
1. 下列各无符号十进制整数中, 能用八位二进制表示的数中最大的是 ()。

- A. 133
- B. 199
- C. 256
- D. 296

2. 下列选项是正确 IP 地址的是 ()。

- A. 100.128.35.91
- B. 111-127-35-21
- C. 192.168.0.3
- D. 202.300.12.4

3. 对图 G 中各个结点分别指定一种颜色, 使相邻结点颜色不同, 则称为图 G 的一个正常着色。正常着色图 G 所必需的最少颜色数, 称为 G 的色数。那么下图的色数是 ()。



- A. 3
- B. 4
- C. 5
- D. 6

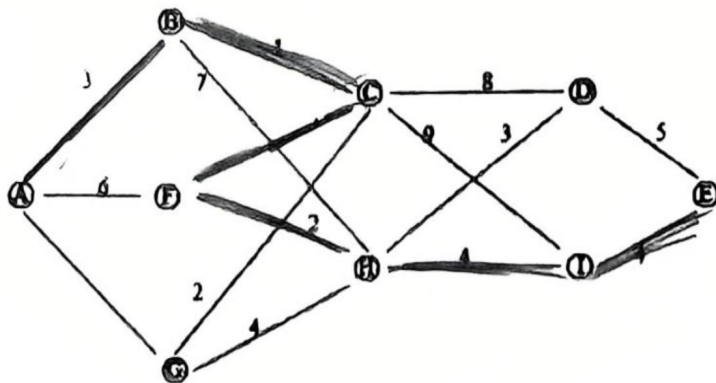
4. 双向链表中有两个指针域, $llink$ 和 $rlink$, 分别指回前驱及后继, 设 p 指向链表中的一个结点, q 指向一待插入结点, 现要求在 p 前插入 q , 则正确的插入为 ()。

- A. $p \rightarrow llink = q; q \rightarrow rlink = p; p \rightarrow llink \rightarrow rlink = q; q \rightarrow llink = p \rightarrow llink;$
- B. $q \rightarrow llink = p \rightarrow llink; p \rightarrow llink \rightarrow rlink = q; q \rightarrow rlink = p; p \rightarrow llink = q \rightarrow rlink;$
- C. $p \rightarrow llink \rightarrow rlink = q; q \rightarrow rlink = p; q \rightarrow llink = p \rightarrow llink; p \rightarrow llink = q;$
- D. $q \rightarrow rlink = p; p \rightarrow rlink = q; p \rightarrow llink \rightarrow rlink = q; q \rightarrow rlink = p;$

5. 前序遍历序列与后序遍历序列相同的二叉树为 ()。

- A. 只有根结点的二叉树
- B. 根结点无右子树的二叉树
- C. 非叶子结点只有左子树的二叉树
- D. 非叶子结点只有右子树的二叉树

6. 设某算法的计算时间表示为递推关系式 $T(n) = T(n-1) + n$, n 为正整数, $T(0) = 1$ 则该算法的时间复杂度为 ()
- A. $\Theta(\log n)$
 B. $\Theta(n \log n)$
 C. $\Theta(n)$
 D. $\Theta(n^2)$
7. 对长度为 n 的有序单链表, 若检索每个元素的概率相等, 则顺序检索到表中任一元素的平均检索长度为 ()。
- A. $\frac{n}{4}$
 B. $\frac{n-1}{2}$
 C. $\frac{n}{2}$
 D. $\frac{n+1}{2}$
8. 线性表若采用链表存储结构, 要求内存中可用存储单元地址 ()。
- A. 一定连续
 B. 一定不连续
 C. 部分地址一定连续
 D. 连续不连续均可
9. 设 G 是有 6 个结点的完全图, 要得到一颗生成树, 需要从 G 中删去 () 条边。
- A. 6
 B. 9
 C. 10
 D. 15
10. 具有 n 个顶点, e 条边的图采用邻接表存储结构, 进行深度优先遍历和广度优先遍历运算的时间复杂度均为 ()。
- A. $\Theta(n + e)$
 B. $\Theta(n^2)$
 C. $\Theta(e^2)$
 D. $\Theta(n^2)$
11. 在数据压缩编码的应用中, 哈夫曼算法是一种采用了 () 思想的算法。。
- A. 分治
 B. 贪心
 C. 递推
 D. 回溯
12. 如图所示, 图中每条边上的数字表示该边的长度, 则从 A 到 E 的最短距离是 ()。



- A. 15
- B. 16
- C. 17
- D. 20

13. 同时查找 $2n$ 个数中的最大值和最小值, 最少比较次数为 ()

- A. $2n - 2$
- B. $3n - 2$
- C. $4n - 2$
- D. $\frac{3(n-2)}{2}$

14. 输入时 n 个不等的数构成的数组 a , 输出 a 中第二小的数。在最坏的情况下, 该算法需要做 () 次比较。

- A. $2n - 1$
- B. $2n - 2$
- C. $2n - 3$
- D. $2n$

```

if (a[1] < a[2])
{
    min1 = a[1];
    min2 = a[2];
}
else
{
    min1 = a[2];
    min2 = a[1];
}
for(int i = 3; i <= n; i++)
    if (a[i] < min2)
        if (a[i] < min1)
        {
            min2 = min1;
            min1 = a[i];
        }
        else
        {
            min2 = a[i];
        }
    }

```

15. 由数字 1, 1, 2, 4, 8, 8 所组成的不同的四位数的个数是 ()

- A. 102
- B. 120
- C. 560
- D. 720

二、阅读程序

- 判断题 1 分, 选择题 3 分, 共计 40 分
- 判断题正确填 T, 错误填 F;

第1题

```
#include<iostream>
int main()
{
    int n, c;
    std::cin >> n >> c;
    long long sum = 0;
    int pre = 1000000000;
    for (int i = 0; i < n; ++i) {
        int a;
        std::cin >> a;
        if (pre > a) {
            pre = a;
        }
        sum += pre;
        pre += c;
    }
    std::cout << sum << "\n";
}
```

判断题

16. 当输入的 a 全都相等时, 程序输出的结果一定等于 $n \times a$ 。
17. 当 $c = 0$ 时, 程序输出的结果一定等于所有 a 的和。

选择题

18. 当 $n = 3$, $c = 0$ 时, 给序列 $a = \{5, 3, 8\}$, 程序输出 ()。
- A. 8
B. 10
C. 11
D. 16
19. 当 $n = 10$, $c = 9$ 时, 给序列 $a = \{3, 1, 4, 15, 9, 26, 53, 58, 97, 9\}$, 程序输出 ()。
- A. 9
B. 90
C. 165
D. 275

第2题

```
bool c[max_size][max_size] = {false};
void draw(int size, int x, int y)
{
    if (size == 1)
    {
        c[x][y] = true;
    }
    else
```

```

{
    int half = size / 2;
    draw(half, x + half, y);
    draw(half, x, y + half);
    draw(half, x + half, y + half);
}
}
void print(int n)
{
    int length = 1 << n;
    draw(length, 0, 0);
    for (int x = 0; x < length; ++x)
    {
        for (int y = 0; y < length; ++y)
        {
            if (c[x][y])
                std::cout << '*';
            else
                std::cout << '.';
        }
        std::cout << "\n";
    }
}

```

判断题

20. 在 draw 过程中, 左上角的四分之一子区域始终不会被绘制。
21. print 输出的星号数量逐行递增。
22. print 输出的星号数量一定比点多。
23. 调用 print(4) 后 c[11][7] 为 true。

选择题

24. print(n) 的时间复杂度为 ()。
 - A. $\Theta(n)$
 - B. $\Theta(2^n)$
 - C. $\Theta(3^n)$
 - D. $\Theta(4^n)$
25. 执行 print(n) 后, 哪个位置一定不是星号 ()。
 - A. (0, 0)
 - B. (length-1, 0)
 - C. (0, length-1)
 - D. (length-1, length-1)
26. 程序输出的星号总数是 ()。
 - A. n
 - B. 2^n
 - C. 3^n
 - D. 4^n

27. 程序输出的图形, 是沿) 对称图形。

- A. 对角线
- B. 水平中轴
- C. 垂直中轴
- D. 中心镜像

第三题

```
bool valid = true;

std::vector<int> adj[max_node];
bool instack[max_node] = {false};
bool visited[max_node] = {false};

void dfs(int node)
{
    instack[node] = true;
    visited[node] = true;
    for (auto after : adj[node]) {
        if (!visited[after]) {
            visited[after] = true;
            dfs(after);
        }
        else if (instack[after]) {
            valid = false;
        }
    }
    instack[node] = false;
}

int main()
{
    int n, m;
    std::cin >> n >> m;
    for (int i = 0; i < m; i++)
    {
        int x, y;
        std::cin >> x >> y;
        adj[x].push_back(y);
    }
    for (int i = 1; i <= n; i++)
    {
        if (!visited[i])
        {
            dfs(i);
        }
    }
    if (valid) {
        std::cout << "Valid\n";
    }
}
```

```

    } else {
        std::cout << "Invalid\n";
    }
}

```

判断题

28. 程序结束时, `instack[]` 数组的所有元素均为 `false`。
29. 程序结束时, `visited[]` 数组的所有元素均为 `true`。
30. 程序在发现图构成一个环后会立即终止 DFS。
31. 若图有多个连通分量, 程序能正确检测所有连通分量中的环。

选择题

32. 若图有 n 个孤立节点 (边数为 0 的图), 程序输出 ()。
- A. valid
- B. Invalid
- C. 都有可能
- D. 编译错误
33. 以下哪种图结构会触发 Invalid 输出 ()。
- A. 树结构
- B. 链式结构
- C. 存在双向边
- D. 完全无环图 (DAG)
34. 当输入节点数 $n = 10^6$, 边数 $m = 2 \times 10^6$ 时, 程序能否高效运行 ()。
- A. 能, $\Theta(n + m)$ 复杂度
- B. 不能, $\Theta(nm)$ 复杂度
- C. 不能, 递归栈溢出
- D. 取决于图的具体结构
35. 以下哪项是环检测的关键判断条件 ()。
- A. `visited[after]`
- B. `!visited[after]`
- C. `instack[after]`
- D. `!instack[after]`

三、完善程序

- 单选题, 每小题 3 分, 共计 30 分

第1题

给定一个 $n \times n$ 的网格。第 $i + 1$ 行、第 $j + 1$ 列的格子 ($0 \leq i, j < n$) 记作 (i, j) 。格子 (i, j) 的颜色由字符 $P[i \bmod N][j \bmod N]$ 决定, 如果是 `B`, 则 (i, j) 是黑格, 如果是 `W`, 则是白格。

给定 Q 个查询, 请依次处理。 每个查询给出 4 个整数 A, B, C, D , 求出以 (A, B) 为左上角、 (C, D) 为右下角的矩形区域内包含的黑格数量。

40. (5)处应填 ()。

- A. $\text{sum}(x2, y2) - \text{sum}(x2, y1 - 1) - \text{sum}(x1 - 1, y2) + \text{sum}(x1 - 1, y1 - 1)$
- B. $\text{sum}(x2, y2) - \text{sum}(x2, y1 + 1) - \text{sum}(x1, y2 + 1) + \text{sum}(x1 - 1, y1 - 1)$
- C. $\text{sum}(x2 - 1, y2 - 1) - \text{sum}(x2 - 1, y1) - \text{sum}(x1, y2 - 1) + \text{sum}(x1, y1)$
- D. $\text{sum}(x2 + 1, y2 + 1) - \text{sum}(x2 + 1, y1) - \text{sum}(x1, y2 + 1) + \text{sum}(x1, y1)$

第2题

给定一个分数 a/b , 若它是一个假分数, 请将它化简成带分数形式输出。例如当 $a/b = 91/30$, 输出

1
3--
30

当输入是一个真分数时, 请将它化简后输出, 且忽略整数部分, 例如当 $a/b = 30/100$ 时, 输出

3
--
10

当输入的分数的可以变成整数时, 忽略它分数部分。注意输出的所有分数都应该是既约的。

```
#include<iostream>

int len(int n) {
    int length = 0;
    while (n > 0) {
        n/=10;
        length++;
    }
    return length;
}

void print(int n, char ch) {
    while (n-->0) std::cout << ch;
}

int main()
{
    int a, b;
    char dummy;
    std::cin >> a >> dummy >> b;
    int p = a;
    int q = b;
    while (a != 0 and b != 0) {
        if (a >= b) a %= b;
        else b %= a;
    }
    int gcd;
    if ( ____ (1) ____ ) gcd = b;
    else gcd = a;
```

```

int i = p / q;
p %= q;
p /= gcd;
q /= gcd;
if ( ____ (2) ____ ) {
    std::cout << i << "\n";
}
else {
    int i_len = len(i);
    int p_len = len(p);
    int q_len = len(q);
    ____ (3) ____ ;
    std::cout << ____ (4) ____ << "\n";
    if (i != 0) std::cout << i ;
    ____ (5) ____ ;
    std::cout << "\n";
    ____ (6) ____ ;
    std::cout << ____ (7) ____ << "\n";
}
}

```

41. (1) 处应填 () 。

- A. a == 0
- B. b == 0
- C. a >= b
- D. b >= a

42. (2) 处应填 () 。

- A. q == p
- B. q % p == 0
- C. q == 0 and p == 1
- D. q == 1 and p == 0

43. (3)(4) 处应填 () 。

- A. print(i_len + q_len - p_len, ' '), p
- B. print(i_len + q_len - p_len, '-') , q
- C. print(i_len + q_len + p_len, ' '), q
- D. print(i_len + q_len + p_len, '-') , p

44. (5) 处应填 () 。

- A. print(i_len, ' ')
- B. print(i_len, '-')
- C. print(q_len, ' ')
- D. print(q_len, '-')

45. (6)(7) 处应填 () 。

- A. print(i_len, ' '), q
- B. print(i_len, '-'), p
- C. print(q_len, ' '), p
- D. print(q_len, '-'), q