

CSP-J 初赛模拟 5

一. 单选题(共 15 题, 每题 2 分, 共计 30 分; 每题有且仅有一个正确选项)

1. 顶级域名.org表示的是(A)
A. 非盈利组织 B. 网络提供商 C. 工商企业 D. 教育机构
2. 一个 64 位长整型变量占用(D)个字节。
A. 32 B. 128 C. 4 D. 8
3. 内存地址的最重要特点是(B)。
A. 随机性 B. 唯一性 C. 顺序性 D. 连续性
4. 一棵最深深度为 n 的二叉树最多和最少结点数分别为多少(D)。
A. $2^n, n$ B. $2^n, n-1$ C. $2^n - 1, n$ D. $2^n - 1, n-1$
5. 最早的计算机的用途是用于(A)
A. 科学计算 B. 自动控制 C. 系统仿真 D. 辅助设计
6. CPU 中(D)机构相当于运算器中的一个存储单元, 它的存取速度比存储器要快得多。
A. 存放器 B. 辅存 C. 主存 D. 寄存器
7. 每份考卷都有一个 8 位二进制序列号。当且仅当一个序列号含有偶数个 1 时, 它才是有效的。例如, 0000000、01010011 都是有效的序列号, 而 11111110 不是。那么, 有效的序列号共有 (C) 个。
A. 64 B. 72 C. 128 D. 164
8. 设栈 S 的初始状态为空, 现有 5 个元素组成的序列 {5, 4, 1, 3, 2}, 对该序列在 S 栈上 依次进行如下操作 (从序列中的第一个元素开始, 出栈后不再进栈): 进栈、进栈、进栈、出栈、出栈、进栈、进栈。试问出栈的元素序列是(D)。
A. {4, 5} B. {4, 1} C. {2, 3} D. {1, 4}
9. 一个箱子中有 7 个红球和 8 个黑球, 取两个球, 都是红球的概率为(B)
A. $1/6$ B. $1/5$ C. $4/15$ D. $8/15$
10. 如果用一个字节来表示整数, 最高位用作符号位, 其他位表示数值。例如: 00000001 表示 +1, 10000001 表示 -1, 试问这样表示法的整数 A 的范围应该是(A)
A. $-127 \leq A \leq 127$ B. $-128 \leq A \leq 128$ C. $-128 \leq A < 128$ D. -128
11. 下列叙述中, 正确的是(D)。
A. 线性表的线性存储结构优于链表存储结构
B. 队列的操作方式是先进后出
C. 栈的操作方式是先进先出
D. 二维数组是指它的每个数据元素为一个线性表的线性表
12. 用某种排序方法对线性表 25, 84, 21, 47, 15, 27, 68, 35, 20 进行排序, 结点变化如下:
(1) 25, 84, 21, 47, 15, 27, 68, 35, 20; (2) 20, 15, 21, 25, 47, 27, 68, 35, 84;
(3) 15, 20, 21, 25, 35, 27, 47, 68, 84; (4) 15, 20, 21, 25, 27, 35, 47, 68, 84.
那么, 排序方法是(D)。
A. 选择排序 B. 希尔排序 C. 合并排序 D. 快速排序
13. 如果某二叉树的前序为 STUWV, 中序为 UWTVS, 那么该二叉树的后序是(A)。

- A. WUVTS B. UWVTS C. VWUTS D. WUTSV

14. 下面关于数据结构的叙述中,正确的叙述是(C)。

- A. 顺序存储方式的优点是存储密度大,且插入、删除运算效率高
B. 链表中的每一个结点都包含一个指针
C. 包含 n 个结点的二叉排序树的最大检索长度为 $\log_2 n$
D. 将一棵树转换为二叉树后,根结点没有右子树

15. 下面哪个排序算法的空间复杂度最大(B) ?

- A. 快速排序 B. 桶排序 C. 归并排序 D. 冒泡排序

二、阅读程序(程序输入不超过数组或字符串定义的范围;判断题正确的填√,错误的填X
除特殊说明外,判断题 1.5 分,选择题 4 分,共计 40 分)

1. 汉诺塔问题:古代有一个梵塔、塔内有三个座 A, B, C, A 座上有 n 个盘子, 盘子大小不等, 大的在下, 小的在上。有一个和尚想把这 n 个盘子从 A 座移到 B 座, 但每次只能允许移动一个盘子, 并且在移动过程中, 3 个座上的盘子始终保持大盘在下, 小盘在上, 程序如下:

```
1  #include<iostream>
2  using namespace std;
3  void hanoi(int n, char a, char b, char c) {
4      if (n == 1)
5          cout<<n<<" "<<a<<" "<<c<<endl;
6      else {
7          hanoi (n-1, a, c, b);
8          cout<<n<<" "<<a<<" "<<c<<endl;
9          hanoi (n-1, b, a, c);
10     }
11 }
12 int main() {
13     int n;
14     cin>>n;
15     hanoi (n, 'A', 'B', 'C');
16     return 0;
17 }
```

●判断题

- (1) (1 分) 当 $n \geq 0$ 时, 程序不会出现死循环。(F)
(2) (1 分) 输出共有 2^n 行。(F)
(3) 当 $n > 0$ 时, 将第 4 行的 “=” 改为 “<=”, 程序输出结果必定不变。(T)
(4) 将第 5 行的 “n” 改为 “1”, 程序输出结果必定不变。(T)

●选择题

- (5) (3 分) 此程序的时间复杂度是(D)。
A. $O(n)$ B. $O(n^2)$ C. $O(n^3)$ D. $O(2^n)$
(6) 若要求输出不超过 15 行, 则下列哪个 n 的值是合法的?(B)
A. 0 B. 4 C. 5 D. 6

```
1  #include<cstdio>
```

```

2  #define N 1005
3
4  using namespace std;
5  int num[N];
6
7  int main() {
8      int a1 = 1, n, x;
9      scanf("%d", &n);
10     num[1] = 1;
11     for(int i = 1; i <= n; ++i) {
12         x = 0;
13         for(int j = 1; j <= a1; ++j) {
14             num[j] = num[j] * 5 + x;
15             x = num[j] / 10;
16             num[j] %= 10;
17         }
18         if(x > 0) num[++a1] = x;
19     }
20     printf("0. ");
21     for(int i = a1; i < n; ++i) {
22         putchar('0');
23     }
24     for(int i = a1; i >= 1; i--) {
25         printf("%d", num[i]);
26     }
27     putchar('\n');
28     return 0;
29 }

```

●判断题

- (1) (1分) 程序输出的是 5^n 的值。(F)
- (2) (1分) 程序执行到第 27 行时, i 的值为 1。(F)
- (3) 对于任意 $1 \leq i \leq a1$, 都有 $0 \leq num[i] \leq 9$ 。(T)
- (4) 程序输出的是一个小数, 且小数末尾可能有多余的 0。(F)

●选择题

- (5) 此程序的时间复杂度是(B)
- A. $O(n)$ B. $O(n^2)$ C. $O(n^3)$ D. $O(n \log n)$
- (6) 若 $n=3$, 则输出为(B)
- A. 8 B. 0.125 C. 0.8 D. 125

3. 在起点和终点之间, 有 N 块岩石(不含起点和终点的岩石)。在比赛过程中, 选手们将从起点出发, 每一步跳向相邻的岩石, 直至到达终点。

为了提高比赛难度, 组委会计划移走一些岩石, 使得选手们在比赛过程中的最短跳跃距离尽可能长。由于预算限制, 组委会至多从起点和终点之间移走 M 块岩石(不能移走起点和终点的岩石)。

输入文件第一行包含三个正整数 L, N, M, 分别表示起点到终点的距离, 起点和终点之间的岩石数, 以及组委会至多移走的岩石数。

接下来 N 行, 每行一个整数, 第 i 行的整数 $a[i]$ ($0 < a[i] < L$) 表示第 i 块岩石与起点的距离。这些岩石按与起点距离从小到大的顺序给出, 且不会有两个岩石出现在同一个位置。

输出文件只包含一个整数, 即最短跳跃距离的最大值。

```
1  #include<iostream>
2  using namespace std;
3  int l, n, m, a[50005], ans;
4  bool check(int dis)
5  {
6      int count = 0, last = 0;
7      for(int i = 1; i <= n; i++)
8          if(a[i]-last<dis) count++;
9          else last = a[i];
10     if(count>m) return 0;return 1;
11 }
12 int main()
13 {
14     ios::sync_with_stdio(0);
15     cin>>l>>n>>m;
16     for(int i = 1; i <= n; i++)
17         cin>>a[i];
18     a[n+1] = l;
19     int fl = 0, fr = l;
20     while(fl<=fr)
21     {
22         int mid = (fl+fr)/2;
23         if(check(mid)) fl = mid+1, ans = mid;
24         else fr = mid-1;
25     }
26     cout<<ans;
27     return 0;
28 }
```

●判断题

- (1) (1 分) 将第 19 行的 “fl=0” 改为 “fl=l”, 程序输出结果必定不变。 (T)
- (2) (2 分) 程序执行到第 26 行时, 必有 $fl > fr$ 。 (T)
- (3) (2 分) 若第 23 行执行的 $check(mid) == 1$, 则最终的 $ans \leq$ 此时的 mid 。 (F)
- (4) (2 分) 程序执行到第 10 行时, $count$ 的值表示: 如果最短跳跃距离恰好为 dis , 那么最少需要移走几块岩石。 (F)

●选择题

- (5) 此程序的时间复杂度是 (C)。
A. $O(n^2)$ B. $O(nl)$ C. $O(n \log l)$ D. $O(n \log n)$
- (6) 若输入为:

25 5 2

2

11

14

17

21

则输出为(B)。

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

三、完善程序(每小题 2 分, 共计 30 分)

1. 完全背包问题。容量为 10 的背包, 有 5 种物品, 每种物品数量无限, 其重量分别为 5, 4, 3, 2, 1, 其价值分别为 1, 2, 3, 4, 5。设计算法, 实现背包内物品价值最大。代码如下(输出 50):

```
1   #include<iostream>
2   #include<algorithm>
3
4   using namespace std;
5
6   int main()
7   {
8       int total_weight = 10;
9       int w[6] = {0, 5, 4, 3, 2, 1};
10      int v[6] = {0, 1, 2, 3, 4, 5};
11      int dp[11] = {__①__};
12
13      for(int i = 1; i <= __②__; i++)
14          for(int j = w[i]; j <= __③__; j++)
15              dp[j] = __④__;
16
17      cout<< __⑤__ << endl;
18      return 0;
19 }
```

(1) ①处应填(A)。

A. 0 B. 5 C. 10 D. 15

(2) ②处应填(A)

A. 5 B. 6 C. 10 D. 15

(3) ③处应填(C)。

A. 5 B. 6 C. 10 D. 15

(4) ④处应填(D)

A. $dp[j]+v[i]$ B. $dp[j-w[i]]+v[i]$
C. $\min(dp[j], dp[j-w[i]]+v[i])$ D. $\max(dp[j], dp[j-w[i]]+v[i])$

(5) ⑤处应填(B)。

A. $v[10]$ B. $dp[10]$ C. $w[10]$ D. $total_weight$

2. (2017 大整数除法) 给定两个正整数 p 和 q , 其中 p 不超过 10^{100} , q 不超过 100000, 求 p 除以 q 的商和余数。输入: 第一行是 p 的位数 n , 第二行是正整数 p , 第三行是正整数 q 。输出: 两行, 分别是 p 除以 q 的商和余数。

```
#include <iostream>
using namespace std;
int p[100];
int n, i, q, rest;
char c;
int main()
{
    cin >> n;
    for (i = 0; i < n; i++)
    {
        cin >> c;
        p[i] = c - '0';
    }
    cin >> q;
    rest = ____ (1);
    i = 1;
    while (____ (2) && i < n)
    {
        rest = rest * 10 + p[i];
        i++;
    }
    if (rest < q)
        cout << 0 << endl;
    else
    {
        cout << ____ (3); ____
        while (i < n)
        {
            rest = ____ (4); ____
            i++;
            cout << rest / q;
        }
        cout << endl;
    }
    cout << ____ (5) << endl;
    return 0;
}
```

● 选择题

(1)①处应填(C)。

A.q[0] B.n C. p[0] D.0

(2)②处应填(A)。

- A. $rest < q$ B. $rest \neq q$ C. $rest \leq q$ D. $rest$
- (3) ③处应填(A)。
- A. $rest/q$ B. $rest \% q$ C. $rest$ D. $rest \& q$
- (4) ④处应填(C)。
- A. $pow(i,10)\%q+p[i]$ B. $rest*10+ p[i]\%q$
 C. $rest\%q*10+p[i]$ D. $rest+p[i]$
- (5) ⑤处应填(D)。
- A. $rest-a$ B. $rest$ C. $rest / q$ D. $rest \% q$