

## 普及组模拟题第八套试题及答案

1. IPv4 中, 以下 IP 地址不合法的是 ( B )

A. 255.255.255.255    B. 0.1.1.1    C. 1.1.1.0    D. 1.0.0.0

【解析】IPv4 中, 合法 IP 地址范围为 (1~255).(0~255).(0~255).(0~255)。所以答案为 B。

2. 已知 A, B, C 是 3 个二进制数, 符号  $\wedge$  表示逻辑与运算, 符号  $\vee$  表示逻辑或运算。若 A=1100

1101 0011    B=1100 0111 0110    C=0011 0110 1010

则表达式 (A $\vee$ B)  $\wedge$  (A $\vee$ C) 的值为 ( C )

A. 1100 1110 0001    B. 0011 0010 1111

C. 1100 1111 0011    D. 1100 0111 0001

【解析】A=1100 1101 0011    B=1100 0111 0110

A $\vee$ B 1100 1111 0111

A=1100 1101 0011    C=0011 0110 1010

A $\vee$ C 1111 1111 1011

与运算

1100 1111 0111

1111 1111 1011

1100 1111 0011

3. Linux 下可执行文件的默认扩展名为 ( D )

A. exe    B. chm    C. dll    D. 都不是

【解析】都是 Windows 的可执行文件。

4. 八进制数 7042 转化为十六进制数是 ( C )

A. 3521    B. F22    C. E22    D. 111000100010

【解析】7042 转二进制

111 000 100 010

再转十六进制

1110 0010 0010

所以结果是 e22

5. 以下排序算法中, 不需要进行关键字比较操作的算法是 ( A )

A. 基数排序    B. 冒泡排序    C. 堆排序    D. 直接插入排序

【解析】基数排序: 不需要进行关键字的比较。

6. 一个袋子中有 3 个蓝球, 2 个红球, 2 个黄球, 则从中抽出三个球颜色各不相同的概率是多少?

( C )

A. 10/21    B. 13/33    C. 12/35    D. 3/7

【解析】考虑三种颜色的球出现的顺序: 红黄蓝、红蓝黄、蓝红黄、蓝黄红、黄蓝红、黄红蓝。对于第一种顺序, 抽到的概率为  $(2/7) * (2/6) * (3/5)$ , 同理求出剩下五种, 求和即可。

7. 10 名学生平均分成两组，每组都围城一个圆圈，有多少种不同的排法（ B ）

A. 30240    B. 72576    C. 72578    D. 30242

**【解析】**

复杂一些的环排问题，第一组有5个元素。10个元素中选5个元素围成一圈，

环排公式： $\frac{1}{m} * A_{10}^5 = \frac{1}{5} * A_{10}^5$ ，剩下一组剩下的5个元素中选5个出来， $A_{n-1}^{n-1} =$

$A_4^4 = 4!$ ，即  $\frac{1}{5} * A_{10}^5 * 4!$ 。还得继续去重。

例如：

第一组 1 2 3 4 5 第二组 6 7 8 9 10 也可以

第二组 6 7 8 9 10 第一组是 1 2 3 4 5，但是这两种情况是重复情况。围成两

个圆圈没有差别，最后再除以  $A_2^2$ ，所以最终结果  $\frac{1}{5} * A_{10}^5 * 4! / A_2^2 = 72576$

8. 定义一颗有根树的深度：根结点的深度为 0，其余结点的深度等于该结点的父亲结点的深度加

1。以下数字中哪一个可以作为一颗深度为 9 的完全二叉树的总节点数？（ C ）

A. 511    B. 1024    C. 1023    D. 1026

**【解析】**

题里的深度为 9 的完全二叉树，实际深度为 10，最多的结点数是  $2^{10}-1$ ，完全二叉树，结点数范围在  $512 \sim 1023$  之间。

9. 共 9 个互不相同的数，它们的最大公约数是 2021 的一个大于 1 的因子（6 有 2、3、6 这三个大于 1 的因子，因子可以包含自身），且这 9 个数的和小于等于 2021，则这 9 个数的和是多少？

（ B ）

A. 1849    B. 1935    C. 2021    D. 1927

**【解析】**

2021 分解为  $43 * 47$ 。抽屉原理：取 43 为最大公约数时，剩余 9 个数的和取 1-9，得到答案 B；取 47 为最大公约数，9 个数必然会有重复。

10. 以下哪位科学家被称为“博弈论之父”，“现代计算机之父”？（ B ）

A. 图灵    B. 冯诺依曼    C. 塔扬    D. 比尔盖茨

**【解析】** 常识题冯·诺依曼是现代电子计算机与博弈论的重要创始人。

11. 设栈 S 和队列 Q 初始状态为空，元素  $a_1, a_2, \dots, a_6$  依次通过栈 S，一个元素出栈后就进入队列 Q，若出队的顺序分别是  $a_2, a_4, a_3, a_6, a_5, a_1$ ，则栈 S 的容量至少是（ B ）

A. 2    B. 3    C. 4    D. 5

**【解析】**  $a_1, a_2$  入栈， $a_2$  出栈， $a_3, a_4$  进栈， $a_4, a_3$  出栈， $a_5, a_6$  进栈， $a_6, a_5, a_1$  出栈，所以栈的容量至少为 3。

12. 对有序数组 {5, 13, 19, 21, 37, 56, 64, 75, 88, 92, 100} 进行二分查找, 等概率的情况下查找成功的平均查找长度 (平均比较次数) 是 ( B )

A. 35/11    B. 34/11    C. 3    D. 32/11

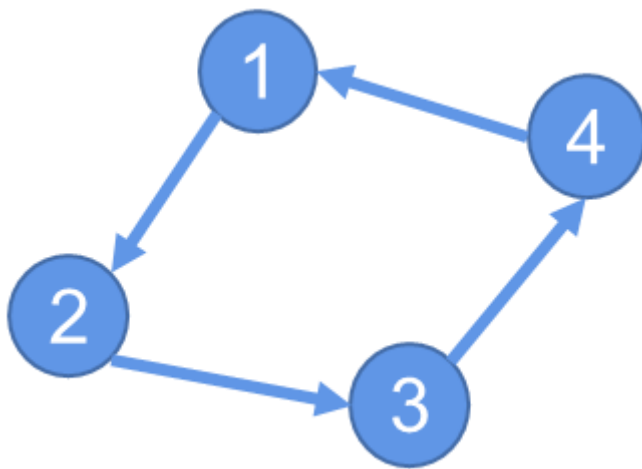
【解析】19, 88 需要查找 2 次; 56 需要查找 1 次; 13, 37, 75 需要查找 3 次; 5, 21, 64, 92, 100 需要查找 4 次。

13. 一个  $n$  个顶点的强连通图最少有几条边 ( A )。

A.  $n$     B.  $n+1$     C.  $n-1$     D.  $n*(n-1)$

【解析】强连通图是指一个有向图中任意两点  $v_1$ 、 $v_2$  间存在  $v_1$  到  $v_2$  的路径及  $v_2$  到  $v_1$  的路径的图。

强连通图的边数最少时为一个环., 所以最少有  $n$  条边。如右图:



14. 在 1 和 2015 之间 (包括 1 和 2015 在内) 不能被 4、5、6 三个数任意一个数整除的数有几个? ( C )

A. 1035    B. 1105    C. 1075    D. 2000

【解析】

容斥原理,  $2015 - 503 - 403 - 335 - 100 + 67 + 167 - 33 = 1075$ 。

15. 关于 Catalan 数  $C_n$ , 下列说法中错误的是 ( A )。

A.  $C_n$  表示有  $n+1$  个结点的不同形态的二叉树的个数。

B.  $C_n$  表示含  $n$  对括号的合法括号序列的个数。

C.  $C_n$  表示长度为  $n$  的入栈序列对应的合法出栈序列个数。

D.  $C_n$  表示通过连接顶点而将  $n+2$  边的凸多边形分成三角形的方法个数。

【解析】

是  $n$  个节点的不同形态二叉树个数。

阅读程序

```
1  #include <bits/stdc++.h>
2  using namespace std;
```

```

3   int p;
4   void fun(int &x, int &y);
5   void func(int &x, int &y) {
6       if(y>x) return;
7       x--;y/=2;
8       fun(x, y);
9   }
10  void fun(int &x, int &y) {
11      if(x==1) return;
12      x/=2;y+=p;
13      func(x, y);
14  }
15  int main() {
16      int x, y;
17      cin>>x>>y>>p;
18      fun(x, y);
19      cout<<x<<' '<<y;
20      return 0;
21}

```

(1) 将第四行的&去除后, 程序仍能通过编译。( )

答案 ×

【解析】四行是函数的声明, 四行 十行必须一致。

(2) 读入的 x, y, p 为 int 范围内任意值时程序均能完成运行。( )

答案 ×

【解析】若 x, y, p 均为-3, 则程序运行永远不会停止。

(3) 若 x=1 时, 输出的 x, y 与输入的一致。( )

答案 ✓

【解析】x=1 时, 直接返回。

(4) 输出的 x 必然小于等于输入的 x。( )

答案 ×

【解析】x 可能为负数, 故此题为错。

(5) 输入为 7 33 2 时, 输出为 ( D )

A. 4 31    B. 4 35    C. 3 31    D. 3 35

【解析】模拟计算得到结果。

(6) 输入为 33 7 2 时, 输出为 ( B )

A. 5 3    B. 3 5    C. 6 4    D. 4 6

【解析】

模拟计算得到结果。

(2)

```

1   #include <iostream>
2   using namespace std;

```

```

3  const int maxn=105;
4  int n,a[maxn],b[maxn];
5  int main()
6  {
7      cin>>n;
8      int x;
9      for(int i=1;i<=n;i++){
10         cin>>x;
11         a[i]=b[i]=x;
12     }
13
14     for(int i=1;i<=n;i++)
15         for(int j=i+1;j<=n;j++){
16             if(a[i]>a[j])swap(a[i],a[j]);
17             if(b[i]<b[j])swap(b[i],b[j]);
18         }
19
20     for(int i=1;i<=n;i++) cout<<a[i]<<" ";
21     cout<<"\n";
22     for(int i=1;i<=n;i++) cout<<b[i]<<" ";
23     cout<<"\n";
24     return 0;
25}

```

(1) 若输入的  $X[1], X[2], \dots, X[N]$  中有相同的数, 程序会陷入死循环。( )

答案 ×

【解析】显然每两个数之间只会被比较一次, 故程序必然结束, 错误。

(2) 当且仅当输入的  $X[1], X[2], \dots, X[N]$  全部相同时输出的两行结果相同。( )

答案 ✓

【解析】a 数组将原序列从小到大排序, b 数组将原序列从大到小排序, 故当且仅当输入的  $X[1], X[2], \dots, X[N]$  全部相同时输出的两行结果相同, 正确。

(3) 该算法的原理是基数排序。( )

答案 ×

【解析】用的选择排序思想。

(4) 若输入的  $X[1], X[2], \dots, X[N]$  互不相同, 则下列说法正确的是 ( B )

- A. 输出的两行结果相同
- B. 将输出的第一行结果整体翻转后, 将与第二行相同
- C. 将输出的第一行结果的第一项与最后一项交换后, 将与第二行相同
- D. 以上说法都不正确

【解析】a 数组将原序列从小到大排序, b 数组将原序列从大到小排序, 故 B 正确。

(5) 下列说法不正确的是 ( D )

- A. 输出的第一行即为将  $X[1], X[2], \dots, X[N]$  从小到大排序后得到的结果
- B. 输出的第二行即为将  $X[1], X[2], \dots, X[N]$  从大到小排序后得到的结果

C. 若将 “ $a[i]>a[j]$ ” 改为 “ $a[i]\geq a[j]$ ”，则程序输出无变化

D. 不存在时间复杂度更优的能与本程序达到相同目的算法

【解析】

a 数组将原序列从小到大排序，b 数组将原序列从大到小排序，故 ABC 正确，且快速排序可以在  $O(N\log N)$  时间内完成这一过程，D 错误。

(6) 该程序的时间复杂度为 ( C )

A.  $O(n)$       B.  $O(n\log n)$       C.  $O(n^2)$       D.  $O(n\sqrt{n})$

【解析】

双重循环，时间复杂度为  $O(N^2)$ 。

(3)

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  int main() {
4      int num=0, sum=1;
5      cin>>num;
6      //保证 num>=100, 且在 int 范围内
7      int max_primedivisor=0;
8      int cnt=1;
9      for(int i=2; i*i<=num; i++) {
10         if(num%i==0) {
11             int tmp=1;
12             while(num%i==0) num/=i, tmp++;
13             max_primedivisor=max(max_primedivisor, i);
14             sum*=tmp;
15         }
16     }
17     max_primedivisor=max(max_primedivisor, num);
18     if(num>1) cnt*=2;
19     cout<<max_primedivisor<<" "<<cnt<<"\n";
20     return 0;
21 }
```

●

代码解析：

算法过程如下：找到一个当前 num 的最小质因子，并将其除干净，直到 num 为一个质数，因此  $\text{max\_primedivisor}=\max(\text{max\_primedivisor}, \text{num})$ ；这句话是有必要的。并且因为最小质因子  $< O(\sqrt{\text{num}})$ ，所以最坏时间是复杂度  $O(\sqrt{\text{num}})$ ，而当  $\text{num}=2^k$  时，时间复杂度最优为  $O(\log \text{ num})$ 。再通过计算或模拟发现 cnt 存的是 num 的约数个数，由此可解出正确答案。

(1) 代码中  $\text{max\_primedivisor}=\max(\text{max\_primedivisor}, \text{num})$ ；这句话去掉对答案没有影响。

( )

答案 ×

【解析】找到一个当前 num 的最小质因子，并将其除干净，直到 num 为一个质数，因此  $\text{max\_primedivisor}=\max(\text{max\_primedivisor}, \text{num})$ ；这句话是有必要的。

(2) 当读入的  $num=p*q$ , 其中  $p<q$ , 且  $p, q$  为质数, 则 for 循环中  $i$  遍历到  $q$  时退出循环。( )

答案 ×

【解析】 $i*i \leq num$ ; 循环到  $\sqrt{num}$ 。

(3) 该算法的最坏时间复杂度为 ( B )

A.  $O(\log num)$       B.  $O(\sqrt{num})$       C.  $O(num)$       D.  $O(num \sqrt{num})$

【解析】因为最小质因子  $< O(\sqrt{num})$ , 所以最坏时间是复杂度  $O(\sqrt{num})$ , 而当  $num=2^k$  时, 时间复杂度最优为  $O(\log num)$ 。

(4) 当读入 2021 时输出为 ( C )

A. 43 2      B. 43      4      C. 47 2      D. 47 4

```
【解析】int num=0, sum=1;
cin>>num;
//保证 num>=100, 且在 int 范围内
int max_primedivisor=0; int cnt=1;
for(int i=2; i*i<=num; i++) { //2021
    if(num%i==0) {
        int tmp=1;
        //43 cout<<i<<endl;
        while(num%i==0) num/=i, tmp++; //47
        max_primedivisor=max(max_primedivisor, i); //43
        sum*=tmp; //1*2
    }
}
max_primedivisor=max(max_primedivisor, num); //47
if(num>1) cnt*=2; //cnt=1*2;
cout<<max_primedivisor<<" "<<cnt<<"\n";
```

模拟一下结果 47 2

(5) 在最好的情况下, 时间复杂度为 ( A )

A.  $O(\log num)$       B.  $O(\sqrt{num})$       C.  $O(num)$       D.  $O(num \sqrt{num})$

【解析】而当  $num=2^k$  时, 时间复杂度最优为  $O(\log num)$ 。

完善程序

1. (电老鼠与方阵) 有一个  $n * n$  ( $2 \leq n \leq 5000$ ) 的方阵, 其中每个方格有一个电力值。小 Y 可以在这个方阵中得到电力, 方法就是在一些方格放上电老鼠来吸收电力, 这样就可以获得这些方格上的电力。不过放的电老鼠须要遵循两个规则: 1. 一个方格最多只能放一只电老鼠; 2. 所有  $2*2$  的子矩阵 (共有  $(n-1)*(n-1)$  个) 必须恰好包含两只电老鼠。小 Y 用了一个程序求出了能获得的最大总电力值。试补全程序。

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 const int N=5100;
4 int a ① ;
5 int main() {
6     int n, ans1=0, ans2=0;
7     scanf("%d", &n);
8     for(int i=1; i<=n; i++)
```

```

9      for(int j=1;j<=n;j++)
10          scanf("%d",&a[i][j]);
11  for(int i=1;i<=n;i++){
12      int odd=0,even=0;
13      for(int j=1;j<=n;j++){
14          int x= ② ;
15          if(j&1) odd+=x;else even+=x;
16      }
17      ans1+=max(odd,even);
18  }
19  for(int i=1;i<=n;i++){
20      int odd=0,even=0;
21      for(int j=1;j<=n;j++){
22          int x= ③ ;
23          if( ④ ) even+=x;else odd+=x;
24      }
25      ans2+=max(odd,even);
26  }
27  printf("%d\n", ⑤ );
28  return 0;
29 }

```

(1) ①处应该填 ( D )

A. [N][2]    B. [2][N]    C. [N][1100]    D. [5100][5100]

**【解析】**

$n \times n$  ( $2 \leq n \leq 5000$ ) 的方阵.

```

8      for(int i=1;i<=n;i++)
9          for(int j=1;j<=n;j++)
10              scanf("%d",&a[i][j]);

```

所以  $a[5100][5100]$ ;

(2) ②处应该填 ( B )

A.  $a[j][i]$     B.  $a[i][j]$     C.  $a[i+j][(i+j)\&1]$     D.  $a[(i+j)\&1][i+j]$

**【解析】** 每行的值赋值给  $x$ 。

(3) ③处应该填 ( A )

A.  $a[j][i]$     B.  $a[i][j]$     C.  $a[i+j][(i+j)\&1]$     D.  $a[(i+j)\&1][i+j]$

**【解析】** 每列的值赋值给  $x$ 。

(4) ④处应该填 ( C )

A.  $j\&1$     B.  $j|1$     C.  $!(j\&1)$     D.  $!(j|1)$

**【解析】** 判断是否是偶数。

(5) ⑤处应该填 ( A )

A.  $\max(ans1, ans2)$     B.  $\min(ans1, ans2)$   
 C.  $ans1+ans2$     D.  $\max(ans1, ans2)-\min(ans1, ans2)$



【解析】求最大值。

2. (排列) 给定一个  $1 \sim n$  的排列 A, 你需要给出一个  $1 \sim n$  的排列 B, 使得排列 B 的字典序的值最小。输出字典序最小的排列 B。

输入两行, 第一行一个正整数 n,

第二行 n 个整数表示排列 A。

提示: 将问题分为 n 是奇数和 n 是偶数考虑, 贪心处理。

试补全程序。

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int A[1000010];int B[1000010];int C[1000010];
4 int main()
5     int n;scanf("%d",&n)
6     for(int i=1;i<=n;i++) scanf("%d",&A[i]);
7
8     if( ① ){
9         int p1=0;int p2= ② ;
10        for(int i=1;i<=n;i++){
11            if(A[i]>n/2){
12                B[i]=++p1
13            }else{
14                B[i]=++p2;
15            }
16        }
17    }else{
18        int p1=0;int p2=n/2;
19        for(int i=1;i<=n;i++){
20            if(A[i]> ③ ){
21                B[i]=++p1;
22            }else{
23                B[i]=++p2;
24            }
25        }
26
27        p1=0;p2=n/2+1;
28        for(int i=1;i<=n;i++){
29            if(A[i]>= ④ ){
30                C[i]=++p1;
31            }else{
32                C[i]=++p2;
33            }
34        }
35
36        int flag=0;
37        for(int i=1;i<=n;i++){
38            if(B[i]<C[i]){flag=1;break;}
39            if(B[i]>C[i]){flag=2;break;}
40        }
41        if(flag!= ⑤ ) swap(B,C);
```

```

42     }
43     for(int i=1;i<n;i++) printf("%d ",B[i]);
44     printf("%d\n",B[n]);
45     return 0;
46 }

```

(1) ①处应填 ( A )

A.  $n\%2==0$     B.  $n\%2==1$     C.  $n==1$     D.  $n==2$

**【解析】**

(以下的  $n/m$  都表示 C++ 中的  $n$  整除  $m$ )

当  $n$  为偶数时, 将  $1 \sim n$  分为  $1 \sim n/2$  (第一类),  $n/2+1 \sim n$  (第二类), 我们使 A 中的第一类匹配 B 中的第二类, 使 A 中的第二类匹配 B 中的第一类, 这样便可使排列 B 的字典序的值最小, 因此我们可以从前往后贪心地放数, 从前往后遍历排列 A, 遇到第一类数, 放当前还没用过的最小的第二类数; 遇到第二类数, 放当前还没用过的最小的第一类数。

所以①处需要先判断  $n$  是否为偶数, 故选 A;

(2) ②处应填 ( C )

A.  $p1$     B.  $n/2-1$     C.  $n/2$     D.  $n/2+1$

**【解析】** ②处的  $p2$  表示的是当前还没用过的第二类数的最小值-1, 所以初始值应该赋  $n/2$ 。

(3) ③处应填 ( D )

A.  $p1$     B.  $n/2-1$     C.  $n/2$     D.  $n/2+1$

**【解析】** 当  $n$  为偶数时, 情况类似, 但由于  $n/2+1$  的存在, 我们需要分两种情况: A 数组中的  $1 \sim n/2$  匹配 B 数组中的  $n/2+2 \sim n$ , B 数组中的  $1 \sim n/2+1$  匹配 A 数组中的  $n/2+1 \sim n$ ; 或 A 数组中的  $1 \sim n/2+1$  匹配 B 数组中的  $n/2+1 \sim n$ , A 数组中的  $n/2+2 \sim n$  匹配 B 数组中的  $1 \sim n/2$  (两种情况的值是相同的)。

所以我们将两种情况分别算一遍, 取字典序较小的即可。

对于③处, 观察到上面的  $p2$  初始值为  $n/2$ , 是上文提到的第二种情况, 所以可与  $p1$  匹配的  $A[i]$  满足  $A[i] > n/2+1$ , 选 D。

(4) ④处应填 ( D )

A.  $p1$     B.  $n/2-1$     C.  $n/2$     D.  $n/2+1$

**【解析】** 则下面是第一种情况, 所以可与  $p1$  匹配的  $A[i]$  满足  $A[i] \geq n/2+1$ , 所以④处填 D;

(5) ⑤处应填 ( B )

A. 0    B. 1    C. 2    D. 3

**【解析】** 对于⑤处, 因为我们最后输出的都是 B 数组, 如果 C 数组的字典序小于 B 数组, 我们必须交换 B 数组和 C 数组, 由上面的循环可知,  $flag=1$  时表示 B 数组的字典序小于 C 数组, 所以  $flag!=1$  交换不影响答案, 故选 B。