

2023 年 12 月认证 C++ 五级真题解析

CCF 编程能力等级认证，英文名 Grade Examination of Software Programming (以下简称 GESP)，由中国计算机学会发起并主办，是为青少年计算机和编程学习者提供学业能力验证的平台。GESP 覆盖中小学全学段，符合条件的青少年均可参加认证。GESP 旨在提升青少年计算机和编程教育水平，推广和普及青少年计算机和编程教育。

GESP 考察语言为图形化 (Scratch) 编程、Python 编程及 C++ 编程，主要考察学生掌握相关编程知识和操作能力，熟悉编程各项基础知识和理论框架，通过设定不同等级的考试目标，让学生具备编程从简单的程序到复杂程序设计的编程能力，为后期专业化编程学习打下良好基础。

本次为大家带来的是 2023 年 12 月份 C++ 五级认证真题解析。

一、单选题（每题 2 分，共 30 分）

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

1、下面 C++ 代码用于求斐波那契数列，该数列第 1、2 项为 1，以后各项均是前两项之和。下面有关说法错误的是()。



```
1 int fiboA(int N)
2 {
3     if (N == 1 || N == 2)
4         return 1;
5     return fiboA(N - 1) + fiboA(N - 2);
6 }
7 int fiboB(int N)
8 {
9     if (N == 1 || N == 2)
10        return 1;
11    int last2 = 1, last1 = 1;
12    int nowVal = 0;
13    for (int i = 2; i < N; i++)
14    {
15        nowVal = last1 + last2;
16        last2 = last1;
17        last1 = nowVal;
18    }
19    return nowVal;
20 }
```

- A. fiboA() 用递归方式，fiboB() 循环方式
- B. fiboA() 更加符合斐波那契数列的数学定义，直观易于理解，而fiboB() 需要将数学定义转换为计算机程序实现
- C. fiboA() 不仅仅更加符合数学定义，直观易于理解，且因代码量较少执行效率更高
- D. fiboB() 虽然代码量有所增加，但其执行效率更高

【答案】C

【考纲知识点】 算法知识点

【解析】 fiboA 是很好理解的，但是执行效率不高，有的计算是重复的，导致效率低。

2、下面C++代码以递归方式实现合并排序，并假设 merge (int T[], int R[], int s, int m, int t) 函数将有序（同样排序规则）的 T[s..m]和 T[m+1..t]归并到 R[s..t]中。横线处应填上代码是()。

```
1 void mergeSort(int SList[], int TList[], int s, int t, int len)
2 {
3     if (s == t){
4         TList[s] = SList[s];
5         return;
6     }
7     int *T2 = new int[len]; // 保存中间结果
8     int m = (s + t) / 2;
9     _____;
10    merge(T2, SList, s, m, t);
11    delete T2;
12    return ;
13 }
```

- A. mergeSort(SList, T2, s, m,len), mergeSort(SList, T2, m,t,len)
- B. mergeSort(SList, T2, s, m-1,len), mergeSort(SList, T2, m+1,t,len)
- C. mergeSort(SList, T2, s, m,len), mergeSort(SList, T2, m+1,t,len)
- D. mergeSort(SList, T2, s, m-1,len), mergeSort(SList, T2, m-1,t,len)

【答案】C

【考纲知识点】算法知识点

【解析】本题考察归并排序。归并排序需要先将排序序列一分为二，左边的元素的区间是[s,m],右边元素区间是[m+1,t]，然后递归排序两个子序列后，将有序的子序列合并。

3、阅读下面的 C++代码 ， 执行后其输出是()。

```
1 int stepCount = 0;
2 int fracA(int N)
3 {
4     stepCount += 1;
5     cout << stepCount << "->";
6     int rtn = 1;
7     for (int i = 1; i <= N; i++)
8         rtn *= i;
9     return rtn;
10 }
11 int fracB(int N)
12 {
13     stepCount += 1;
14     cout << stepCount << "->";
15     if (N == 1)
16         return 1;
17     return N * fracB(N - 1);
18 }
19 int main()
20 {
21     cout << fracA(5);
22     cout << "<====>";
23     cout << fracB(5);
24     return 0;
25 }
```

- A. 1->120<====>2->120
- B. 1->120<====>1->120
- C. 1->120<====>1->2->3->4->5->120
- D. 1->120<====>2->3->4->5->6->120

【答案】D

【考纲知识点】算法知识点

【解析】本题考察递归算法。输出 `fracA` 函数，是先输出 1，再输出 5 的阶乘，120；23 行代码，执行 `fracB` 函数，此时 `stepCount` 从 2 开始计数，依次输出 2/3/4/5/6，再输出 5 的阶乘 120。

4、下面的 C++ 用于对 `lstA` 排序，使得偶数在前奇数在后，横线处应填入()。

```
1 bool isEven(int N)
2 {
3     return N % 2 == 0;
4 }
5
6 void swap(int &a, int &b)
7 {
8     int t;
9     t=a,a=b,b=t;
10    return;
11 }
12
13 void sortA(int lstA[], int n)
14 {
15     int i,j,t;
16     for (i = n-1; i > 0; i--)
17         for(j = 0; j < i; j++)
18             if(_____)
19                 swap(lstA[j], lstA[j+1]);
20
21     return;
22 }
```

- A. `isEven(lstA[j]) && !isEven(lstA[j+1])`
- B. `!isEven(lstA[j]) && isEven(lstA[j+1])`
- C. `lstA[j] > lstA[j+1]`
- D. `lstA[j] < lstA[j+1]`

【答案】A

【考纲知识点】排序算法知识点

【解析】本题考察排序算法。前一个数字，下标是 `j` 的数字是偶数，后面的数字下标是 `j+1` 的是奇数，按照要求，偶数在奇数的后面，要交换。A 符合题意条件。

5、下面的 C++ 代码用于将字符串保存到带头节点的双向链表中，并对重复的串计数，然后将最新访问的串节点放在链头便于查找。横线处应填入代码是()。



```
1 typedef struct Node{
2     string str;
3     int ref;
4     struct Node *next, *prev;
5 }Node;
6 Node * Insert(Node *pHead, string s)
7 {
8     Node *p = pHead->next;
9     Node *q;
10    while(p){
11        if(p->str == s){
12            p->ref++;
13            p->next->prev = p->prev;
14            p->prev->next = p->next;
15            break;
16        }
17        p=p->next;
18    }
19    if(!p) {
20        p = new Node;
21        p->str = s;
22        p->ref=0;
23        p->next = p->prev = NULL;
24    }
25
26    pHead->next = p, p->prev = pHead;
27    return pHead;
28 }
```

- A. if(pHead) {p->next = pHead->next, pHead->next->prev = p;}
- B. if(pHead->next) {p->next = pHead->next, pHead->next->prev = p;}
- C. p->next = pHead->next, pHead->next->prev = p;
- D. 触发异常，不能对空指针进行操作。

【答案】 B

【考纲知识点】 指针知识

【解析】本题考察双链表知识点。每个节点需要 2 个指针，指向前驱节点和后继节点。按照要求，新的节点要求插入到链表头部。头节点和新插入的节点都需要修改。B 选项能够完成新节点的插入。

6、有关下面C++代码说法正确的是（ ）。

```
1 int rc;
2 int foo(int x, int y)
3 {
4     int r;
5     if(y == 0)
6         r = x;
7     else{
8         r = foo(y, x % y);
9         rc++;
10    }
11    return r;
12 }
```

- A. 如果 x 小于 10， rc 值也不会超过 20

- B. foo 可能无限递归
- C. foo 可以求出 x 和 y 的最大公共质因子
- D. foo 能够求出 x 和 y 的最小公倍数

【答案】 A

【考纲知识点】 数学知识点

【解析】 本题考察数学算法，求最大公约数。这是典型的最大公约数写法的变形。排除法选 A。

7、下面的 C++ 代码实现对 list 的快速排序，有关说法，错误的是（ ）。

```
1 vector<int> operator +(vector<int>lA,vector<int>lB)
2 {
3     vector<int>lst;
4
5     for (int i = 1; i < lA.size(); i++)
6         lst.push_back(lA[i]);
7     for (int i = 1; i < lB.size(); i++)
8         lst.push_back(lB[i]);
9
10    return lst;
11 }
12
13 vector<int>qSort (vector<int>lst)
14 {
15     if (lst.size() < 2)
16         return lst;
17     int pivot = lst[0];
18     vector<int>less, greater;
19     for (int i = 1; i < lst.size(); i++)
20         if (lst[i] <= pivot) less.push_back(lst[i]);
21         else greater.push_back(lst[i]);
22     for (int i = 1; i < lst.size(); i++)
23         if (lst[i] <= pivot) less.push_back(lst[i]);
24         else greater.push_back(lst[i]);
25
26     return _____;
27 }
```

- A. `qSort(less) + qSort(greater) + (vector<int>)pivot`
- B. `(vector<int>)pivot + (qSort(less) + qSort(greater))`
- C. `(qSort(less) + (vector<int>)pivot + qSort(greater))`
- D. `qSort(less) + pivot + qSort(greater)`

【答案】 C

【考纲知识点】 排序算法知识点

【解析】 本题考察快速排序。Less 数组保存的是小于等于 pivot，然后加上 pivot 元素，再加上大于等于 pivot 的数组。

8、下面C++代码中的 `isPrimeA()` 和 `isPrimeB()` 都用于判断参数 `N` 是否素数，有关其时间复杂度的正确说法是（ ）。

```
1 bool isPrimeA(int N)
2 {
3     if (N < 2)
4         return false;
5     for (int i = 2; i <= N / 2; i++)
6         if (N % i == 0)
7             return false;
8     return true;
9 }
10 bool isPrimeB(int N)
11 {
12     if (N < 2)
13         return false;
14     for (int i = 2; i <= sqrt(N); i++)
15         if (N % i == 0)
16             return false;
17     return true;
18 }
```

- A. `isPrimeA()` 的最坏时间复杂度是 $O(\frac{N}{2})$ ，`isPrimeB()` 的最坏时间复杂度是 $O(\log N)$ ，`isPrimeA()` 优于 `isPrimeB()`
- B. `isPrimeA()` 的最坏时间复杂度是 $O(\frac{N}{2})$ ，`isPrimeB()` 的最坏时间复杂度是 $O(N^{\frac{1}{2}})$ ，`isPrimeB()` 绝大多数情况下优于 `isPrimeA()`
- C. `isPrimeA()` 的最坏时间复杂度是 $O(N^{\frac{1}{2}})$ ，`isPrimeB()` 的最坏时间复杂度是 $O(N)$ ，`isPrimeA()` 优于 `isPrimeB()`
- D. `isPrimeA()` 的最坏时间复杂度是 $O(\log N)$ ，`isPrimeB()` 的最坏时间复杂度是 $O(N)$ ，`isPrimeA()` 优于 `isPrimeB()`

【答案】 B

【考纲知识点】 数学知识点

【解析】 本题考察数学知识，判断质数。A 函数时间复杂度是 $O(n/2)$ ，B 函数算法是 $O(\sqrt{n})$ ，大部分情况后者是优的，值更小。

9、下面C++代码用于有序 list 的二分查找，有关说法错误的是（ ）。



```
1 int _binarySearch(vector<int>lst, int Low, int High, int Target)
2 {
3     if (Low > High)
4         return -1;
5     int Mid = (Low + High) / 2;
6     if (Target == lst[Mid])
7         return Mid;
8     else if (Target < lst[Mid])
9         return _binarySearch(lst, Low, Mid - 1, Target);
10    else
11        return _binarySearch(lst, Mid + 1, High, Target);
12}
13 int bSearch(vector<int>lst, int Val)
14 {
15     return _binarySearch(lst, 0, lst.size(), Val);
16 }
```

- A. 代码采用二分法实现有序 list 的查找
- B. 代码采用分治算法实现有序 list 的查找
- C. 代码采用递归方式实现有序 list 的查找
- D. 代码采用动态规划算法实现有序 list 的查找

【答案】 D

【考纲知识点】 算法知识点

【解析】 本题考察算法知识点。二分法每次规模减半，查找平均时间复杂度是 B。

10、在上题的_binarySearch 算法中，如果 lst 中有 N 个元素，其时间复杂度是()。

- A. $O(N)$
- B. $O(\log N)$
- C. $O(N \log N)$
- D. $O(N^2)$

【答案】 B

【考纲知识点】 算法知识点

【解析】 本题考察算法知识点。二分法每次规模减半，单词查找平均时间复杂度是 B。

11、下面的 C++ 代码使用数组模拟整数加法，可以处理超出大整数范围的加法运算。横线处应填入代码是 () 。



```
1 vector<int> operator +(vector<int> a, vector<int> b)
2 {
3     vector<int> c;
4     int t = 0;
5
6     for(int i = 0; i < a.size() || i < b.size(); i ++ )
7     {
8         if(i < a.size()) t = t + a[i];
9         if(i < b.size()) t = t + b[i];
10
11     }
12
13     if(t) c.push_back(t);
14
15     return c;
16 }
```

- A. c.push_back(t % 10), t = t % 10;
- B. c.push_back(t / 10), t = t % 10;
- C. c.push_back(t / 10), t = t / 10;
- D. c.push_back(t % 10), t = t / 10;

【答案】 D

【考纲知识点】 算法知识点

【解析】 本题考察高精度知识点。每次保存对应位和的最低位数字，去掉最低位数字后，保持进位，循环执行。

12、有关下面C++代码的说法正确的是（ ）。

```
1 class Node
2 {
3 public:
4     int Value;
5     Node* Prev;
6     Node* Next;
7     Node(int Val, Node* Prv = NULL, Node* Nxt = NULL);
8 };
9
10 Node::Node(int Val, Node* Prv, Node* Nxt)
11 {
12     this->Value = Val;
13     this->Prev = Prv;
14     this->Next = Nxt;
15 }
16
17 int main()
18 {
19     Node firstNode = Node(10);
20     firstNode.Next = new Node(100, &firstNode);
21     firstNode.Next->Next = new Node(111, firstNode.Next);
22 }
```

- A. 上述代码构成单向链表
- B. 上述代码构成双向链表
- C. 上述代码构成循环链表

D. 上述代码构成指针链表

【答案】 B

【考纲知识点】 链表知识点

【解析】 本题考察链表知识点。每个节点指向自己前一个节点和后一个节点，因此是双向链表。

13、通讯卫星在通信网络系统中主要起到（ ）的作用。

- A. 信息过滤
- B. 信号中继
- C. 避免攻击
- D. 数据加密

【答案】 B

【考纲知识点】 计算机基础知识

【解析】 本题考察计算机基础知识。通信卫星可以转发无线电信号，实现通信地球站间或地球站与航天器间的无线电通信，因此具有信号中继作用。选 B。

14、小杨想编写一个判断任意输入的整数 N 是否为素数的程序，下面哪个方法不合适？（ ）

- A. 埃氏筛法
- B. 线性筛法
- C. 二分答案
- D. 枚举法

【答案】 C

【考纲知识点】 数学知识

【解析】 本题考察数学知识。线筛和埃筛都可以判断素数，枚举也可以，二分规模减半，不能合理判断。

15、下面的排序算法都要处理多趟数据，哪种排序算法不能保证在下一趟处理时从待处理数据中选出最大或最小的数据？（ ）

- A. 选择排序
- B. 快速排序
- C. 堆排序
- D. 冒泡排序

【答案】 B

【考纲知识点】 排序算法知识

【解析】 本题考察排序算法知识。需要了解每种排序算法的特点。快速排序是选定一个数字，每次把比它小的放在左边，比元素大的放在右边，不能确定最值。

二、判断题（每题 2 分，共 20 分）

题号	1	2	3	4	5	6	7	8	9	10
答案	√	×	×	√	√	√	×	√	√	√

1、归并排序的时间复杂度是 $O(N \log N)$ 。()

【答案】 正确

【考纲知识点】 排序算法知识

【解析】 本题考察排序算法知识。归并排序算法的时间复杂度的描述正确。

2、小杨在生日聚会时拿一块 $H*W$ 的巧克力招待来的 K 个小朋友，保证每位小朋友至少能获得一块相同大小的巧克力。那么小杨想分出来最大边长的巧克力可以使用二分法。()

【答案】 错误

【考纲知识点】 算法知识

【解析】 因为考纲中对二分法同时列出了“二分查找”和“二分答案（或二分枚举）”。

3、以下 C++ 代码能以递归方式实现斐波那契数列，该数列第 1、2 项为 1，以后各项均是前两项之和。()



```
1 int Fibo(int N)
2 {
3     if (N == 1 || N == 2)
4         return 1;
5     else
6     {
7         int m = fiboA(N - 1);
8         int n = fiboB(N - 2);
9         return m + n;
10    }
11 }
```

【答案】 错误

【考纲知识点】 算法知识

【解析】 本题考察递归算法知识。递归函数要调用自己。

4、贪心算法可以达到局部最优，但可能不是全局最优解。()

【答案】 正确

【考纲知识点】 算法知识

【解析】 本题考察贪心算法知识。贪心是局部达到最优。

5、小杨设计了一个拆数程序，它能够将任意的非质数自然数 N 转换成若干个质数的乘积，这个程序是可以设计出来的。()

【答案】 正确

【考纲知识点】 数学知识

【解析】 本题考察数学知识。素数分解定理规定:任何一个整数都可以被分解为一系列因子的乘积,乘积中所有的因子都是质数(即素数)。(更严谨一点:大于 1 的整数)

6、插入排序有时比快速排序时间复杂度更低。()

【答案】 正确

【考纲知识点】 排序知识

【解析】 本题考察排序算法知识。当数据初始有序时，插入排序的最快时间复杂度是 $O(n)$ ，快排最坏时间复杂度是 $O(N^2)$ 。

7、下面的 C++代码能实现十进制正整数 N 转换为八进制并输出。()



```
1 char s[10];
2 int main()
3 {
4     int N;
5     cin >> N;
6     string rst = "";
7     while (N != 0)
8     {
9         s[0]=N % 8 + '0';
10        rst += string(s);
11        N /= 8;
12    }
13    cout << rst << endl;
14
15    return 0;
16 }
```

【答案】 错误

【考纲知识点】 计算机基础知识

【解析】 本题考察进制转换知识。转换后的内容要倒序输出并以 0 开头。

8、对数组 `int arr[] = {2, 6, 3, 5, 4, 8, 1, 0, 9, 10}` 执行 `sort(arr, arr+10)`，则执行后 `arr` 中的数据调整为 `{0, 1, 2, 3, 4, 5, 6, 8, 9, 10}`。 ()

【答案】 正确

【考纲知识点】 排序知识

【解析】 本题考察排序算法知识。`sort` 默认是从小到大排序。

9、小杨想写一个程序来算出正整数 `N` 有多少个因数，经过思考他写出了一个重复没有超过 `N/2` 次的循环就能够算出来了。 ()

【答案】 正确

【考纲知识点】 数学知识

【解析】 本题考察数学知识。可以循环 `N` 的一半找到所有因数。

10、同样的整数序列分别保存在单链表和双向链中，这两种链表上的简单冒泡排序的复杂度相同。 ()

【答案】 正确

【考纲知识点】 排序算法知识

【解析】 本题考察排序算法知识。冒泡排序，相邻的数据交换，而且修改节点链的操作不会改变复杂度。

三、编程题（每题 25 分，共 50 分）

题号	1	2
答案		

1、小杨的幸运数

问题描述

小杨认为，所有大于等于 a 的完全平方数都是他的超级幸运数。

小杨还认为，所有超级幸运数的倍数都是他的幸运数。自然地，小杨的所有超级幸运数也都是幸运数。

对于一个非幸运数，小杨规定，可以将它一直 $+1$ ，直到它变成一个幸运数。我们把这个过程叫做幸运化。例如，如果 $a=4$ ，那么 4 是最小的幸运数，而 1 不是，但我们可以连续对 1 做 3 次 $+1$ 操作，使其变为 4 ，所以我们可以说， 1 幸运化后的结果是 4 。

现在，小样给出 N 个数，请你首先判断它们是不是幸运数；接着，对于非幸运数，请你将它们幸运化。

输入描述

第一行 2 个正整数 a, N 。

接下来 N 行，每行一个正整数 x ，表示需要判断（幸运化）的数。

输出描述

输出 N 行，对于每个给定的 x ，如果它是幸运数，请输出 `lucky`，否则请输出将其幸运化后的结果。

特别提醒

在常规程序中，输入、输出时提供提示是好习惯。但在本场考试中，由于系统限定，请不要在输入、输出中附带任何提示信息。

样例输入 1

```
1 2 4
2 1
3 4
4 5
5 9
```

样例输出 1

```
1 4
2 lucky
3 8
4 lucky
```

样例解释 1

1 虽然是完全平方数，但它小于 a ，因此它并不是超级幸运数，也不是幸运数。

将其进行3次+1操作后，最终得到幸运数 4。

4 是幸运数，因此直接输 lucky。

5 不是幸运数，将其进行3次+1操作后，最终得到幸运数 8。

9 是幸运数，因此直接输出 lucky。

样例输入 2

```
1 16 11
2 1
3 2
4 4
5 8
6 16
7 32
8 64
9 128
10 256
11 512
12 1024
```

样例输出 2

```
1 16
2 16
3 16
4 16
5 lucky
6 lucky
7 lucky
8 lucky
9 lucky
10 lucky
11 lucky
```

数据规模

对于 30%的测试点，保证 $a, x \leq 100$ ， $N \leq 100$ 。

对于 60%的测试点，保证 $a, x \leq 10^6$

对于所有测试点，保证 $a \leq 1,000,001$ ；保证 $N \leq 2 \times 10^5$ ，保证 $1 \leq x \leq 1,000,001$

【题目大意】 给一个完全平方数的标准，推导出哪些是超级幸运数，然后判断 N 个数字中，哪些是超级幸运数，是的输出“lucky”，不是的输出离该数字最近的比它大的数字。

【考纲知识点】 数学知识，埃筛知识，循环知识

【解题思路】 完全平方数按照定义，包括 $1,4,9,16,25\cdots$ 。超级幸运数还可以是完全平方数的倍数，因此 $8,12,18\cdots$ 也是超级幸运数。数据范围比较大， $1e6$ ，用埃筛的模板，判断每一个数字是否是幸运数字。不是幸运数字的话，保留离它最近的幸运数字作为答案。最终完成整个查询。

【参考程序】

```
1  #include <cstdio>
2  #include <cstdlib>
3  #include <cstring>
4  #include <algorithm>
5  #include <string>
6  #include <map>
7  #include <iostream>
8
9  #include <cmath>
10 using namespace std;
11 const int N = 1001 * 1001;
12 const double eps = 1e-8;
13 bool is_lucky[N + 5];
14 int next_lucky[N + 5];
15 void init() {
16 }
17 int main() {
18     int a, T;
19     scanf("%d%d", &a, &T);
20     for (int i = 1; i <= N; i++) {
21         int t = int(sqrt(i) + eps);
22         if (i >= a && t * t == i)
23             is_lucky[i] = 1;
24         if (! is_lucky[i])
25             continue ;
26
27         for (int j = i + i; j <= N; j += i)
28             is_lucky[j] = 1;
29     }
30     for(int i = N; i; i --)
31         next_lucky[i] = is_lucky[i] ? i : next_lucky[i + 1];
32
33     while(T --) {
34         int x;
35         scanf("%d", &x);
36         if (is_lucky[x])
37             cout << "lucky" << endl;
38         else
39             cout << next_lucky[x] << endl;
40     }
41     return 0;
42 }
```

2、烹饪问题

问题描述

有 N 种食材，编号从 0 至 $N - 1$ ，其中第 i 种食材的美味度为 a_i 。

不同食材之间的组合可能产生奇妙的化学反应。具体来说，如果两种食材的美味度分别为 x 和 y ，那么它们的契合度为 $x \text{ and } y$ 。其中， and 运算为按位与运算，需要先将两个运算数转换为二进制，然后在高位补足 0 ，再逐位进行与运算。例如， 12 与 6 的二进制表示分别为 1100 和 0110 ，将它们逐位进行与运算，得到 0100 ，转换为十进制得到 4 ，因此 $12 \text{ and } 6 = 4$ 。在 **C++** 或 **Python** 中，可以直接使用 $\&$ 运算符表示与运算。

现在，请你找到契合度最高的两种食材，并输出它们的契合度。

输入描述

第一行一个整数 N ，表示食材的种数。

接下来一行 N 个用空格隔开的整数，依次为 a_0, \dots, a_{N-1} ，表示各种食材的美味度。

输出描述

输出一行一个整数，表示最高的契合度。

特别提醒

在常规程序中，输入、输出时提供提示是好习惯。但在本场考试中，由于系统限定，请不要在输入、输出中附带任何提示信息。

样例输入 1

```
1 3
2 1 2 3
```

样例输出 1

```
1 2
```

样例解释 1

可以编号为 $1, 2$ 的食材之间的契合度为 $2 \text{ and } 3 = 2$ ，是所有食材两两之间最高的契合度。

样例输入 2

1	5
2	5 6 2 10 13

样例输出 2

1	8
---	---

样例解释 1

可以编号为 3, 4 的食材之间的契合度为 $10 \text{ and } 13 = 8$ ，是所有食材两两之间最高的契合度。

数据规模

对于 40% 的测试点，保证 $N \leq 1,000$ ；

对于所有测试点，保证 $N \leq 10^6$ ， $0 \leq a_i \leq 2,147,483,647$ 。

【题目大意】 选出 2 个数字，求出这 2 个数字与操作的最大结果是多少。

【考纲知识点】 位运算知识，循环知识，排序知识

【解题思路】 需要选取 2 个数字，可以用双重循环枚举这 2 个数字，取最大值，最终得到答案。对于前 40% 的测试点是可以的。当数据量大的时候，就超时了。我们知道，两个数字对应的二进制，位数越高是 1，越有可能是答案。所以从最高位统计，是否至少有 2 个数字最高位是 1，保存最高位结果，并且把最高位非 1 的数字删去；再查询次高位是否至少有 2 个以上的数字二进制位都是 1，以此类推，求出哪 2 个数字与 结果最大。利用快速排序的方法每次将剩余数字中当前考虑的位为 1 的数字放在前面，时间复杂度是 $O(N \cdot \log(a_i \text{ 值的最大值}))$ 。

【参考程序】



GESP

```
1 #include<cstdio>
2 #include<iostream>
3 #include<algorithm>
4 #include<cstdlib>
5 #include<cstring>
6 using namespace std;
7
8 const int MAX_N = int(1e6) + 100;
9
10 int a[MAX_N];
11
12 int sort(int l,int r,int k) {
13     while(l <= r) {
14         while ((l <= r) && (a[l] >> k & 1)) l++;
15         while ((l <= r) && (!(a[r] >> k & 1))) r--;
16         if (l <= r) swap(a[l++], a[r--]);
17     }
18     return r;
19 }
20
21 int main() {
22     int n, j, ans=0;
23     scanf("%d", &n);
24     for (int i = 1; i <= n; i++) scanf("%d", &a[i]);
25     for (int i = 31; i >= 0; i--)
26         if ((j = sort(1, n, i)) >= 2) {
27             ans = ans | 1 << i;
28             n = j;
29         }
30     printf("%d\n", ans);
31     return 0;
32 }
```



CCF-GESP编程能力等级认证
Grade Examination of Software Programming

GESP 2024年3月认证
认证语言: C++/Python/图形化编程
报名时间: 2024年1月18日至3月5日24点截止
缴费时间: 2024年1月18日至3月5日24点截止
认证时间: 1-4级 2024年3月16日 上午 09:30-11:30
5-8级 2024年3月16日 下午 13:30-16:30
认证方式: 全国统一线下机考

扫码即刻报名

GESP面向全国征集授权服务中心和考点 (考点仅限公立校)
欢迎申请, 扫码至官网了解更多

【联系我们】

1. GESP 微信: 关注“CCF GESP”公众号, 将问题以文字方式留言即可得到回复。
2. GESP 邮箱: gesp@ccf.org.cn

注: 请在邮件中详细描述咨询的问题并留下考生的联系方式及姓名、身份证号, 以便及时有

效处理。

3. GESP 电话：0512-67656856

咨询时间：周一至周五(法定节假日除外)：上午 8:30-12:00；下午 13:00-17:30

GESP 第五期认证报名已启动，扫描下方二维码，关注 GESP 公众号即可报名

