

CSP – J2019年初赛试题及详细解析

视频讲解

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. 中国的国家顶级域名是 ()

A. `.cn` B. `.ch` C. `.chn` D. `.china`

答案: A

考点: 计算机基础-计算机网络-域名

解析: 典型的国家顶级域名有 `.cn` (中国)、`.us` (美国)、`.uk` (英国)、`.jp` (日本)、`.sg` (新加坡) 等。

典型的通用顶级域名有 `.edu` (教育机构)、`.gov` (政府部门)、`.net` (网络组织)、`.com` (商业组织)、`.org` (非营机构)、`.mil` (军事部门) 等。

2. 二进制数 `11 1011 1001 0111` 和 `01 0110 1110 1011` 进行逻辑与运算的结果是 ()。

```
11 1011 1001 0111
&01 0110 1110 1011
-----
01 0010 1000 0011
```

答案: D

考点: 计算机基础-数制与编码-二进制-位运算

解析: 逐位进行与运算。对于每一位, 0与0得0, 1与0得0, 0与1得0, 1与1得1。

3. 一个32位整型变量占用 () 个字节。

A. 32 B. 128 C. 4 D. 8

答案: C

考点: 计算机基础-计算机体系结构-内存空间

解析: 8位是1字节, 因此32位是4字节。在C++语言中, `int`是最常用的带符号32位整型变量, 可表示数值: $[-2^{31}, 2^{31} - 1]$, `unsigned int`是最常用的无符号32位整型变量, 可表示数值 $[0, 2^{32} - 1]$ 。

4. 若有如下程序段, 其中s、a、b、c均已定义为整型变量, 且a、c均已赋值 (c大于0)

```
s = a;  
for (b = 1; b <= c; b++) s = s - 1;
```

则与上述程序段功能等价的赋值语句是 ()

A. $s = a - c$; B. $s = a - b$; C. $s = s - c$; D. $s = b - c$;

答案: A

考点: 程序设计基础 C++ 语法基础-循环语句

解析: s 初始化为 a , 紧接着 for 循环 c 次, 每次 s 减 1, 因此该程序段相当于 $s = a - c$ 。

5. 设有 100 个已排好序的数据元素, 采用折半查找时, 最大比较次数为 ()

A. 7 B. 10 C. 6 D. 8

答案: A

考点: 程序设计基础-算法与数据结构-折半查找 (二分查找)

解析: 对 100 个有序元素进行折半查找, 每次查找可将检索范围缩小一半。由 $2^6 - 1 < 100 < 2^7 - 1$ 可知, 最大比较次数为 7。

6. 链表不具有的特点是 ()

A. 插入删除不需要移动元素 B. 不必事先估计存储空间

C. 所需空间与线性表长度成正比 D. 可随机访问任一元素

答案: D 考点: 程序设计基础-算法与数据结构-链表

解析: 链表是通过记录每个元素的后继位置来实现数据存储, 所需空间与元素个数成正比, 优点是不必事先估计存储空间、插入或删除指定位置元素的时间复杂度为 $O(1)$; 但缺点是由于其元素的内存地址不连续, 无法进行 $O(1)$ 的随机访问。

7. 把 8 个同样的球放在 5 个同样的袋子里, 允许有的袋子空着不放, 问共有多少种不同的分法? ()

提示: 如果 8 个球都放在一个袋子里, 无论是哪个袋子, 都只算同一种分法

A. 22 B. 24 C. 18 D. 20

答案: C

考点: 数学-计数问题

解析: **枚举法** 求解: 8 个同样的球

- 分 1 个袋子共 1 种方案
 - (8)
- 分 2 个袋子共 4 种方案

答案: C

考点: 数学-公约数与公倍数

解析: 使用辗转相除法可得 $\gcd(319, 377) = \gcd(319, 58) = \gcd(58, 29) = 29$ 。或将两数分解质因数后, 提取公共部分亦可求解。

11. 新学期开学了, 小胖想减肥, 健身教练给小胖制定了两个训练方案。

方案一: 每次连续跑3公里可以消耗300千卡(耗时半小时); 方案二: 每次连续跑5公里可以消耗600千卡(耗时1小时)。

小胖每周周一到周四能抽出半小时跑步, 周五到周日能抽出一小时跑步。另外, 教练建议小胖每周最多跑21公里, 否则会损伤膝盖。请问如果小胖想严格执行教练的训练方案, 并且不想损伤膝盖, 每周最多通过跑步消耗多少千卡? ()

A. 3000 B. 2500 C. 2400 D. 2520

答案: C

考点: 程序设计基础-算法与数据结构-枚举算法

解析: 设方案1执行 x 天, 方案2执行 y 天, 则有 $3x + 5y \leq 21$ $x \leq 4$, $y \leq 3$ 。要求 $300x + 600y$ 的最大值, 枚举可得最优方案为 $x = 2$ 、 $y = 3$, 此时 $300x + 600y$ 为2400。或使用线性规划亦可求解。

贪心: 在 y 尽量大的情况下去尝试, 应该价值最大。

$y = 3$ 时, $x = 2$, 此时 $300x + 600y = 300 \times 2 + 600 \times 3 = 2400$

12. 一副纸牌除掉大小王有52张牌, 四种花色, 每种花色13张。假设从这52张牌中随机抽取13张纸牌, 则至少 () 张牌的花色一致。

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

答案: A

考点: 数学-抽屉原理

解析: 最坏情况, 13张牌对应四种花色的牌数为3、3、3、4。

尽量均匀分布, 才能使得各个花色数量尽量一致, 使最多的不太突出

13. 一些数字可以颠倒过来看, 例如0、1、8颠倒过来还是本身, 6颠倒过来是

9, 9颠倒过来看还是6, 其他数字颠倒过来都不构成数字。类似的, 一些多位数也可以颠倒过来看, 比如106颠倒过来是901。假设某个城市的车牌只由5位数字组成, 每一位都可以取0到9。请问这个城市最多有多少个车牌倒过来恰好还是原来的车牌? ()

A. 60 B. 125 C. 75 D. 100

答案: C

考点：数学-乘法原理

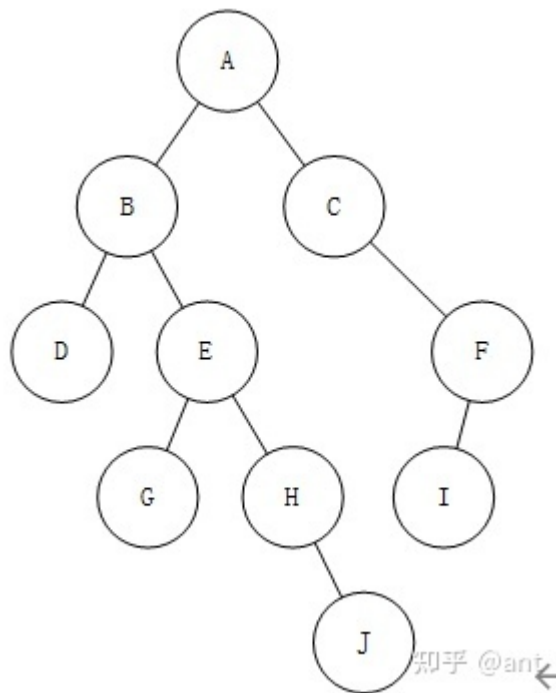
解析：前2位有0、1、8、6、9共5种选择，第3位只能放0、1、8,后2位由前2位决定，因此总方案数为
\$553=75\$。

好题！不这样训练乘法原理，就不知道这东西怎么考查！

14. 假设一棵二叉树的后序遍历序列为 $DGJHEBIFCA$ ，中序遍历序列为 $DBGEHJACIF$ ，则其前序遍历序列为（ ）。

A. $ABCDEFGHIJ$ B. $ABDEGHJCFI$ C. $ABDEGJHCFI$ D. $ABDEGHJFIC$

答案:B



解析：后序遍历的规则是**左右根**、中序遍历的规则是**左根右**，因此可知， A 是树根、 $DBGEHJ$ 是 A 左子树的中序遍历（对应后续遍历 $DGJHEB$ ）、 CIF 是 A 右子树的中序遍历（对应后续遍历 IFC ），递归画出对应的二叉树，再根据前序遍历规则**根左右**即可求出答案。

根据后序遍历和中序遍历求前序遍历，是高频考点，需要训练： [练习I](#) [练习II](#) [练习III](#) [练习IV](#)
[已知前序后序为什么无法求出唯一中序](#)

15. 以下哪个奖项是计算机科学领域的最高奖？（ ）

A. 图灵奖 B. 鲁班奖 C. 诺贝尔奖 D. 普利策奖

答案：A

考点：计算机基础-常识-重要人物

解析：图灵奖由美国计算机协会于1966年设立，其名称取自计算机科学之父图灵，专门奖励对计算机事业作出重要贡献的个人，被誉为“**计算机界的诺贝尔奖**”。

二、阅读程序

```
1  #include <stdio>↵
2  #include <string>↵
3  using namespace std;↵
4  char st[100];↵
5  int main() {↵
6      scanf("%s", st);↵
7      int n = strlen(st);↵
8      for (int i = 1; i <= n; ++i) {↵
9          if (n % i == 0) {↵
10             char c = st[i - 1];↵
11             if (c >= 'a')↵
12                 st[i - 1] = c - 'a' + 'A';↵
13             }↵
14         }↵
15         printf(st);↵
16         return 0;↵
17     }↵
```

知乎 @ant

考点：程序设计基础-算法与数据结构-字符串

概述：程序用于将字符串下标(如果从1开始编号，但C++语言中实际是从0开始编号)是n约数的对应小写字母转换为大写字母。

判断题

1) 输入的字符串只能由小写字母或大写字母组成。 ()

答案：错

解析：输入的字符串也可以包含数字等其他字符。

2) 若将第8行的 $i = 1$ 改为 $i = 0$ ，程序运行时会发生错误。 ()

答案：对

解析：若 i 可以为0,则第9行的if语句条件 $n \% i == 0$ 将发生运行时错误RE。

3) 若将第8行的 $i \leq n$ 改为 $i * i \leq n$, 程序运行结果不会改变。 ()

答案：错

解析：当第8行的循环条件为 $i \leq n$ 时，字符串的末尾字符会被程序加工，但若改为 $i * i \leq n$ ，字符串的末尾字符将不会被程序加工（除非字符串长度为1）。注意是开根号，这不对，比如14位，质因子分解为 2×7 ,7这个质因子就是大于 $\sqrt{14}$,如果只枚举到 $\sqrt{14}$,就会有一个大因子没有遍历到。

4) 若输入的字符串全部由大写字母组成，那么输出的字符串就跟输入的字符串一样。 ()

答案：对

解析：大写字母的ASCII编码值小于小写字母的。若输入的字符串全部由大写字母组成，则程序不会对其进行加工。

•选择题

5) 若输入的字符串长度为18，那么输入的字符串跟输出的字符串相比，至多有 () 个字符不同。

A. 18 B. 6 C. 10 D. 1

答案：B

解析：18的正约数共有6个，因此程序至多修改输入字符串中的6个字符，即输出字符串与输入字符串至多有6个字符不同。

6) 若输入的字符串长度为 ()，那么输入的字符串跟输出的字符串相比，至多有36个字符不同。

A. 36 B. 100000 C. 1 D. 128

答案:B

解析：根据程序的作用可知，要使输出字符串和输入字符串之间至多有36个字符不同，36应当是字符串长度 n 的约数个数。本题选项中，仅有100000满足要求，将其分解质因数得 $100000 = 2^5 \times 5^5$ ，根据约数个数定理，得其的正约数共有 $(5+1)(5+1) = 36$ 个。

程序二

```
1  #include <stdio>
2  using namespace std;
3  int n, m;

4  int a[100], b[100];

6  int main() {
7      scanf("%d%d", &n, &m);
8      for (int i = 1; i <= n; ++i)
9          a[i] = b[i] = 0;
10     for (int i = 1; i <= m; ++i) {
11         int x, y;
12         scanf("%d%d", &x, &y);
13         if (a[x] < y && b[y] < x) {
14             if (a[x] > 0)
15                 b[a[x]] = 0;
16             if (b[y] > 0)
17                 a[b[y]] = 0;
18             a[x] = y;
19             b[y] = x;
20         }
21     }
22     int ans = 0;
23     for (int i = 1; i <= n; ++i) {
24         if (a[i] == 0)
25             ++ans;
26         if (b[i] == 0)
27             ++ans;
28     }
29     printf("%d\n", ans);
30     return 0;
31 }
```

知乎 @ant

假设输入的 n 和 m 都是正整数， x 和 y 都是在 $[1, n]$ 的范围内的整数，完成下面的判断题和单选题：

写在前面,方法总结

- 由于没有专门训练过阅读程序这样逆向思维的试题，导致阅读起来理解有困难，需要总结技巧。

- 不要上来就试图读懂整个代码的逻辑流程，应该从边界入手，从实例入手，从分支入手，去验证问题，不一定非得完整的把代码搞清楚它的原本含义。一般常见的例子举3行，比如：

```
1 2  
2 3  
3 4
```

- 数字一般从1开始，能简单就尽量简单。
- 不要指望一组模拟数据就能过掉所有问题，可能需要两组或以上

考点：程序设计基础-算法与数据结构-模拟算法 此题见17.xls的表格模拟测试数据解法。

程序三

```

1  #include <iostream>
2  using namespace std;
3  const int maxn = 10000;
4  int n;
5  int a[maxn];
6  int b[maxn];
7  int f(int l, int r, int depth) {
8      if (l > r)
9          return 0;
10     int min = maxn, mink;
11     for (int i = l; i <= r; ++i) {
12         if (min > a[i]) {
13             min = a[i];
14             mink = i;
15         }
16     }
17     int lres = f(l, mink - 1, depth + 1);
18     int rres = f(mink + 1, r, depth + 1);
19     return lres + rres + depth * b[mink];
20 }
21 int main() {
22     cin >> n;
23     for (int i = 0; i < n; ++i)
24         cin >> a[i];
25     for (int i = 0; i < n; ++i)
26         cin >> b[i];
27     cout << f(0, n - 1, 1) << endl;
28     return 0;
29 }

```

知乎 @ant

满二叉树和完全二叉树

退化成链的二叉树

考点：程序设计基础-算法与数据结构-树结构

概述：程序可以视作根据二叉树的**中序遍历**(根左右)，构造一棵满足要求的树，并输出各结点深度与**b**值的加权和——要求二叉树的根结点 **最小**(*min*)，并递归要求左右子树的根结点最小(除非相应子树为空)。

判断题

1) 如果 a 数组有重复的数字, 则程序运行时会发生错误。 ()

答案: 错

解析: 若 a 数组有重复数字, 则程序在根据 a 数组递归构造符合要求的二叉树时, 对于相同结点值, **会优先考虑位于左侧的**。

2) 如果 b 数组全为0, 则输出为0。 ()

答案: 对

解析: 程序最终输出的是各结点深度与 b 值的 **加权和**, 因此若 b 数组全为0, 则加权和显然为0。

选择题

3) 当 $n = 100$ 时, **最坏情况下**, 与第12行的比较运算执行的次数最接近的是 ()。

A. 5000 B. 600 C. 6 D. 100

答案: A

解析: 最坏情况下, 程序所构造的二叉树的每个结点至多仅有一个子结点, 此时, 程序将递归100层, 其中第 i 层进行 $100 - i + 1$ 次第12行的比较运算, 总执行次数为 $100 + 99 + 98 + \dots + 1 \approx 5000$ 。

4) 当 $n = 100$ 时, **最好情况下**, 与第12行的比较运算执行的次数最接近的是 ()。

A. 100 B. 6 C. 5000 D. 600

答案: D

解析: 最佳情况下, 程序构造二叉树时, 对于每个结点会尽可能均分其左右子树。定义根结点深度为1, 则含 $n = 100$ 个结点的树的深度最小为 $\log n \approx 7$, 此时每选定一层结点, 程序都需要执行约 n 次的第12行的比较运算, 因此总执行次数约为 $n \log n \approx 600$ 。

5) 当 $n = 10$ 时, 若 b 数组满足, 对任意 $0 \leq i < n$ 。都有 $b[i] = i + 1$, 那么输出最大为 ()。

A. 386 B. 383 C. 384 D. 385

答案: D

解析: 构造数据 $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

会发现, 这是一棵从左上到右下的链, 深度为10

此时, 要使输出的 ans 值尽可能大, 程序所构造的二叉树的深度应尽可能的大。定义根结点深度为1, 则含10个结点的二叉树的最大深度为10, 因此 ans 的最大值为 $1+2+3+\dots+10=55$ 。

6) (4分) 当 $n = 100$ 时, 若 b 数组满足, 对任意 $0 \leq i < n$, 都有 $b[i] = 1$, 那么输出最小为 ()。

A. 582 B. 580 C. 579 D. 581

答案: B

解析：此时，要使输出的 ans 值尽可能小，程序应参照完全二叉树构造此树，其中深度为1的结点共1个，深度为2的结点共2个，深度为3的结点共4个.....深度为6的结点共32个，剩余37个结点的深度为7，因此 ans 的最小值为 $(11+22+34+\dots+632)+7*37=580$ 。

三、完善程序 (单选题，每题3分，共计30分)

视频讲解

程序一

1. (矩阵变幻)有一个奇幻的矩阵，在不停的变幻，其变幻方式为：数字 0 变成矩阵

$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ ，数字 1 变成矩阵 $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ ，最初该知阵只有一个元素 0，变幻 n 次后，矩阵会变成什么样？

例如，矩阵最初为：[0]；矩阵变幻 1 次后： $\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ ；矩阵变幻 2 次后：

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

输入一行一个不超过 10 的正整数 n 。输出变幻 n 次后的矩阵。

试补全程序。

提示：

“<<”表示二进制左移运算符，例如 $(11)_2 \ll 2 = (1100)_2$ ；

而“^”表示二进制异或运算符，它将两个参与运算的数中的每个对应的二进制位一一进行比较，若两个二进制位相同，则运算结果的对应二进制位为 0，反之为 1。

知乎 @ant

```

1  #include <stdio>
2  using namespace std;
3  int n;
4  const int max_size = 1 << 10;

6  int res[max_size][max_size];

8  void recursive(int x,int y,int n,int t) {
9      if (n == 0) {
10         res[x][y]= ① ;
11         return;
12     }
13     int step = 1 << (n - 1);
14     recursive(②,n - 1, t);
15     recursive(x, y + step, n - 1, t);
16     recursive(x + step, y, n - 1, t);
17     recursive(③, n - 1, !t);
18 }
19
20 int main() {
21     scanf("%d", &n);
22     recursive(0,0,④);
23     int size = ⑤ ;
24     for (int i = 0; i < size; ++i) {
25         for (int j = 0; j < size; ++j)
26             printf("%d", res[i][j]);
27         puts("");
28     }
29     return 0;
30 }

```

知乎 @ant

考点：程序设计基础-算法与数据结构-分治算法

概述：程序采用分治算法，递归模拟矩阵的变换过程。递归函数 `recursive(x, y, n, t)` 表示计算左上角 (x, y) ，大小 $2^n * 2^n$ 由单个数字 t 变幻而来的矩阵。

1. ①处应填 ()

A. n B. 0 C. t D. 1

答案: *C*

解析: 此处为递归边界, 当需要计算的是单位矩阵时, 相应元素应赋值为 t , 即无需再经任何变换。

2) ②处应填 ()

A. $x - \text{step}$, $y - \text{step}$ B. x , $y - \text{step}$ C. $x - \text{step}$, y D. x , y

答案: *D*

解析: 左上角 (x, y) , 且大小 $2^n \times 2^n$ 的矩阵, 可以分成 4 个 $2^{n-1} \times 2^{n-1}$ 的矩阵分别计算。此处需要计算的是 4 个矩阵中位于左上方的矩阵, 该矩阵的左上角坐标为 (x, y) 。

3) ③处应填 ()

A. $x - \text{step}$, $y - \text{step}$ B. $x + \text{step}$, $y + \text{step}$ C. $x - \text{step}$, y D. x , $y - \text{step}$

答案: *B*

解析: 左上角 (x, y) , 且大小 $2^n \times 2^n$ 的矩阵, 可以分成 4 个 $2^{n-1} \times 2^{n-1}$ 的矩阵分别计算。此处需要计算的是 4 个矩阵中位于右下方的矩阵, 该矩阵的左上角坐标为 $(x+2^{n-1}, y+2^{n-1})$ 。

4) ④处应填 ()

A. $n - 1$, $n \% 2$ B. n , 0

C. n , $n \% 2$ D. $n - 1$, 0

答案: *B*

解析: 此处是递归计算的入口, 即题目最终所求的是大小 $2^n \times 2^n$, 由单个数字 0 变幻而来的矩阵, 因此递归函数的后两个参数应设为 n 和 0。

5) ⑤处应填 ()

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

答案: *B*

解析: 此处是计算最终所求的矩阵大小, 即边长 $size$ 为 2^n , 位运算写做 $1 \ll n$

程序二

2. (计数排序) 计数排序是一个广泛使用的排序方法。下面的程序使用双关键字计数排序，将 n 对 10000 以内的整数，从小到大排序。

例如有三对整数 (3, 4)、(2, 4)、(3, 3)，那么排序之后应该是 (2, 4)、(3, 3)、(3, 4)。

输入第一行为 n ，接下来 n 行，第 i 行有两个数 $a[i]$ 和 $b[i]$ ，分别表示第 i 对整数的第一关键字和第二关键字。

从小到大排序后输出。

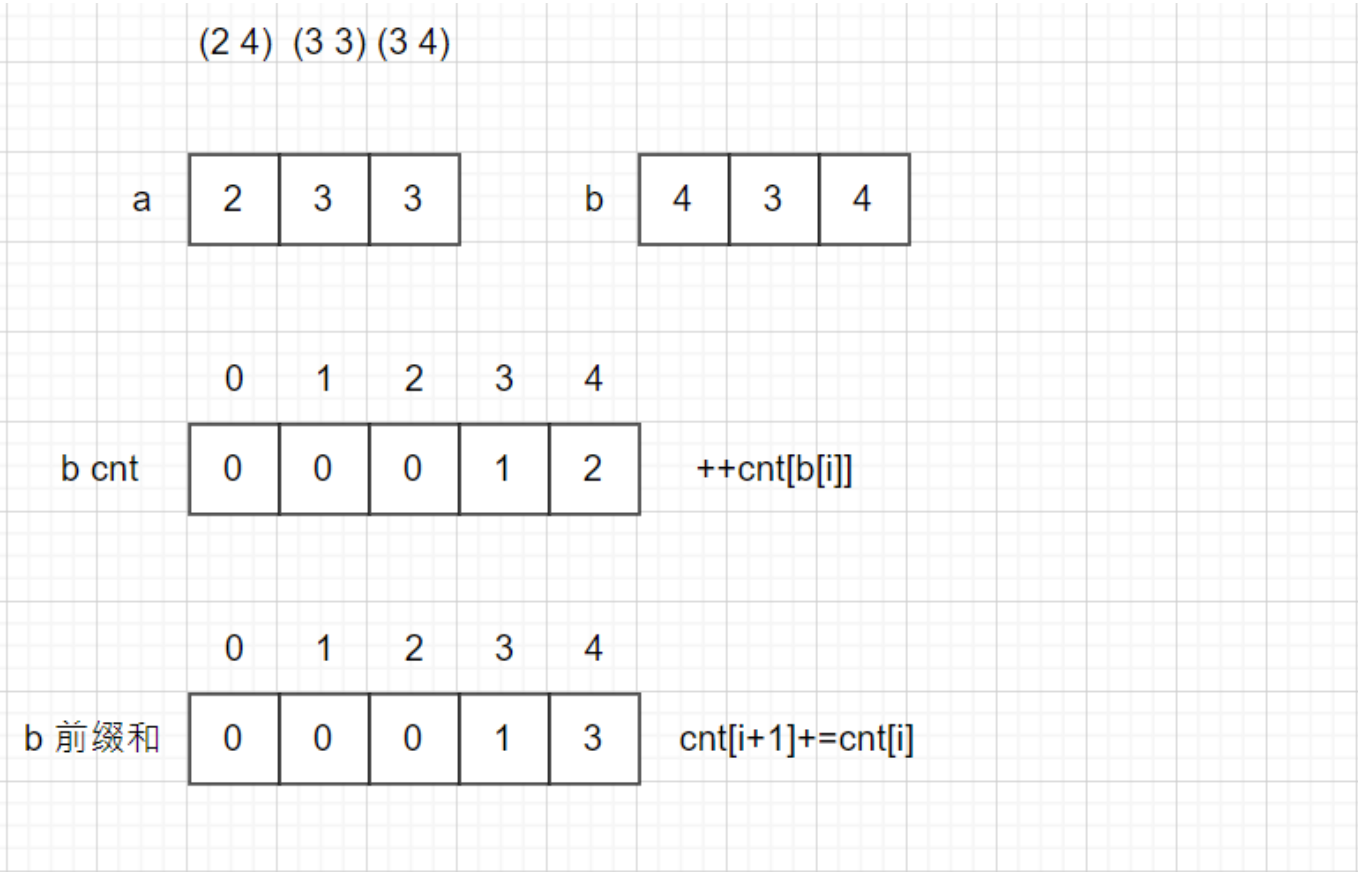
数据范围 $1 \leq n \leq 10^7$ ， $1 \leq a[i], b[i] \leq 10^4$ 。

提示：应先对第二关键字排序，再对第一关键字排序。数组 $ord[]$ 存储第二关键字排序的结果，数组 $res[]$ 存储双关键字排序的结果。
试补全程序。

知乎 @ant

```
1  #include <stdio>↵
2  #include <string>↵
3  using namespace std;↵
4  const int maxn = 10000000;↵
5  const int maxs = 10000;↵
↵
7  int n;↵
8  unsigned a[maxn], b[maxn], res[maxn], ord[maxn];↵
9  unsigned cnt[maxs + 1];↵
10 ↵
11 int main() {↵
12     scanf("%d",&n);↵
13     for (int i = 0; i < n; ++i)↵
14         scanf("%d%d", &a[i], &b[i]);↵
15     memset(cnt,0,sizeof(cnt));↵
16     for (int i = 0; i < n; ++i)↵
17         ① ; //利用 cnt 数组统计数量↵
18     for (int i = 0; i < maxs; ++i)↵
19         cnt[i + 1] += cnt[i];↵
20     for (int i = 0; i < n; ++i)↵
21         ② ; //记录初步排序结果↵
22     memset(cnt, 0 , sizeof(cnt));↵
23     for (int i = 0; i < n; ++i)↵
24         ③ ; //利用 cnt 数组统计数量↵
25     for (int i = 0; i < maxs; ++i)↵
26         cnt[i + 1] += cnt[i];↵
27     for (int i = n - 1; i >= 0; --i)↵
28         ④ ; //记录最终排序结果↵
29     for (int i = 0; i < n; ++i)↵
30         printf("%d %d\n", ⑤); ↵
31     return 0;↵
```

知乎 @ant



考点：程序设计基础-算法与数据结构-排序算法

解析：基于计数排序，程序实际实现了近似于基数排序的算法——依次以低位到高位的一位数为关键词，进行计数排序，前一次的排序结果是下一次的初始序列——本题对应着先根据第二关键词*b*进行计数排序，再根据第一关键词*a*进行计数排序。

1. ①处应填()

A. ++cnt[i] B. ++cnt[b[i]] C. ++cnt[a[i] * maxs + b[i]] D. ++cnt[a[i]]

答案：B

解析：此处是根据第二关键词*b*进行计数排序，并做各关键词的数量统计工作， 因此将*b[i]*对应的元素数量自增1。

2) ②处应填 ()

A. ord[--cnt[a[i]]] = i B. ord[--cnt[b[i]]] = a[i] C. ord[--cnt[a[i]]] = b[i] D. ord[--cnt[b[i]]] = i

答案：D

解析:此处是根据第二关键词*b*进行计数排序，并记录排序结果。此时的*ent[key]* 用于表示关键词为*key*的元素在结果数组中的位置，因此这里的程序应将关键词为*b[i]*的元素*i*放在*ord*数组里。

3) ③处应填 ()

A. ++cnt[b[i]] B. ++cnt[a[i] * maxs + b[i]] C. ++cnt[a[i]] D. ++cnt [i]

答案:C

解析：此处是根据第一关键词 a 进行计数排序，并做各关键词的数量统计工作，因此将 $a[i]$ 对应的元素数量自增1。

4) ④处应填 ()

A. $res[--cnt[a[ord[i]]]] = ord[i]$ B. $res[--cnt[b[ord[i]]]] = ord[i]$ C. $res[--cnt[b[i]]] = ord[i]$ D. $res[--cnt[a[i]]] = ord[i]$

答案：A

解析：此处是根据第一关键词 a 进行计数排序，并记录排序结果。由于此前已经根据第二关键词 b 进行计数排序，此时第 i 个元素的原始下标实际为 $ord[i]$ ，因此这里的程序应将关键词为 $a[ord[i]]$ 的元素 $ord[i]$ 放在 res 数组里。

5) ⑤处应填 ()

A. $a[i], b[i]$ B. $a[res[i]], b[res[i]]$ C. $a[ord[res[i]]], b[ord[res[i]]]$ D. $a[res[ord[i]]], b[res[ord[i]]]$

答案：B

解析：此处是按顺序输出排序结果。由于此前已经按照第二关键词、第一关键词完成了计数排序，此时第 i 个元素的原始下标实际为 $res[i]$ 。

[C++实现排序算法之基数排序](#)