第 1 章　NOIP 普及组(CSP-J)历年试题分析

1.1　2020 CCF 入门级(CSP-J)试题 B 卷及解析

1.1.1　2020 CCF 入门级(CSP-J)试题

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. 在内存储器中每个存储单元都被赋予一个唯一的序号,称为(　　)。

A. 下标　　B. 地址　　C. 序号　　D. 编号

2. 编译器的主要功能是(　　)。

A. 将源程序翻译成机器指令代码　　B. 将一种高级语言翻译成另一种高级语言

C. 将源程序重新组合　　D. 将低级语言翻译成高级语言

3. 设 x=true,y=true,z=false,以下逻辑运算表达式值为真的是(　　)。

A. (x∧y)∧z　　B. x∧(z∨y)∧z

C. (x∧y)∨(z∨x)　　D. (y∨z)∧x∧z

4. 现有一张分辨率为 2048×1024 像素的 32 位真彩色图像。请问要存储这张图像,需要多大的存储空间?(　　)

A. 4MB　　B. 8MB　　C. 32MB　　D. 16MB

5. 冒泡排序算法的伪代码如下。

输入:数组 L,n≥1。

输出:按非递减顺序排序的 L。

算法 BubbleSort:

```
1.  FLAG←n      //标记被交换的最后元素位置
2.  while FLAG > 1 do
3.      k ← FLAG - 1
4.      FLAG ← 1
5.      for j = 1 to k do
6.          if L(j)> L(j + 1)then do
7.          L(j)<->L(j + 1)
8.          FLAG ← j
```

对 n 个数用以上冒泡排序算法进行排序,最少需要比较多少次?(　　)

A. n　　B. n−2　　C. n^2　　D. n−1

6. 设 A 是 n 个实数的数组,考虑下面的递归算法:

```
    XYZ(A[1..n])
1.  if n = 1 then return A[1]
```

```
2.  else temp ← XYZ(A[1..n-1])
3.  if temp < A[n]
4.  then return temp
5.  else return A[n]
```

请问算法 XYZ 的输出是什么？(　　)

A. A 数组的平均　　B. A 数组的最小值

C. A 数组的最大值　　D. A 数组的中值

7. 链表不具有的特点是(　　)。

A. 插入删除不需要移动元素　　B. 可随机访问任一元素

C. 不必事先估计存储空间　　D. 所需空间与线性表长度成正比

8. 有 10 个顶点的无向图至少应该有(　　)条边才能确保是一个连通图。

A. 10　　B. 12　　C. 9　　D. 11

9. 二进制数 1011 转换为十进制数是(　　)。

A. 10　　B. 13　　C. 11　　D. 12

10. 5 个小朋友并排站成一列，其中有两个小朋友是双胞胎。如果要求这两个双胞胎必须相邻，则有(　　)种不同的排列方法。

A. 24　　B. 36　　C. 72　　D. 48

11. 下面所使用的数据结构是(　　)。

压入 A 成[A]　压入 B 成[B,A]　弹出 B 成[A]　压入 C 成[C,A]

A. 哈希表　　B. 二叉树　　C. 栈　　D. 队列

12. 独根树的高度为 1。具有 61 个结点的完全二叉树的高度为(　　)。

A. 7　　B. 5　　C. 8　　D. 6

13. 干支纪年法是中国传统的纪年方法，由 10 个天干和 12 个地支组合成 60 个天干地支。由公历年份可以根据以下公式和表格换算出对应的天干地支。

天干＝(公历年份)除以 10 所得余数

地支＝(公历年份)除以 12 所得余数

天干	甲	乙	丙	丁	戊	己	庚	辛	壬	癸
	4	5	6	7	8	9	0	1	2	3
地支	子	丑	寅	卯	辰	巳	午	未	申	酉	戌	亥
	4	5	6	7	8	9	10	11	0	1	2	3

例如，今年是 2020 年，2020 除以 10 余数为 0，查表为“庚”；2020 除以 12 余数为 4，查表为“子”，所以今年是庚子年。

请问 1949 年的天干地支是(　　)。

A. 己亥　　B. 己丑　　C. 己卯　　D. 己酉

14. 10 个三好学生名额分配到 7 个班级，每个班级至少有一个名额，一共有(　　)种不同的分配方案。

A. 56　　B. 84　　C. 72　　D. 504

15. 有 5 副不同颜色的手套(共 10 只手套，每副手套左右手各 1 只)，一次性从中取 6 只手套，请问恰好能配成两副手套的不同取法有(　　)种。

A. 30　　B. 150　　C. 180　　D. 120

二、阅读程序(程序输入不超过数组或字符串定义的范围；判断题正确填√；错误填×；除特殊说明外，判断题每题 1.5 分，单选题每题 3 分，共计 40 分)

1.

```
#include "cstdlib"
#include "iostream"
using namespace std;

char encoder[26] = {'C', 'S', 'P', 0};
char decoder[26];

string st;

int main(){
        int k = 0;
        for (int i = 0;i < 26;++i)
            if (encoder[i]! = 0)++k;
        for (char x = 'A';x < = 'Z';++x){
            bool flag = true;
            for (int i = 0;i < 26;++i)
                if (encoder[i] == x){
                    flag = false;
                    break;
                }
                if (flag){
                    encoder[k] = x;
                    ++k;
                }
            }
            for (int i = 0;i < 26;++i)
                decoder[encoder[i] - 'A'] = i + 'A';
             cin >> st;
             for (int i = 0;i < st.length();++i)
                st[i] = decoder[st[i] - 'A'];
            cout << st;
            return 0;
    }
```

判断题

(1) 输入的字符串应当只由大写字母组成，否则在访问数组时可能越界。(　　)

(2) 若输入的字符串不是空串，则输入的字符串与输出的字符串一定不一样。(　　)

(3) 将第 12 行的“i＜26”改为“i＜16”，程序运行结果不会改变。(　　)

(4) 将第 26 行的“i＜26”改为“i＜16”，程序运行结果不会改变。(　　)

单选题

(5) 若输出的字符串为“ABCABCABCA”，则下列说法正确的是(　　)。

A. 输入的字符串中既有 A 又有 P　　B. 输入的字符串中既有 S 又有 B

C. 输入的字符串中既有 S 又有 P　　D. 输入的字符串中既有 A 又有 B

(6) 若输出的字符串为“CSPCSPCSPCSP”，则下列说法正确的是(　　)。

A. 输入的字符串中既有 J 又有 R　　B. 输入的字符串中既有 P 又有 K

C. 输入的字符串中既有 J 又有 K　　D. 输入的字符串中既有 P 又有 R

2.

```
#include "iostream"
using namespace std;

long long n, ans;
int k,len;
long long d[ 1000000];

int main(){
        cin >> n >> k;
        d[0] = 0;
        len = 1;
        ans = 0;
        for (long long i = 0;i < n;++i){
            ++d[0];
            for (int j = 0;j + 1 < len;++j){
                if (d[j] == k){
                    d[j] = 0;
                    d[j + 1] += 1;
                    ++ans;
                }
            }
            if (d[len - 1] == k){
                d[len - 1] = 0;
                d[len] = 1;
                ++len;
                ++ans;
        }
    }
    cout << ans << endl;
    return 0;
}
```

假设输入的 n 是不超过 2^{62} 的正整数,k 都是不超过 10000 的正整数,完成下面的判断题和单选题。

判断题

(1) 若 k=1,则输出 ans 时,len=n。(　　)

(2) 若 k>1,则输出 ans 时,len 一定小于 n。(　　)

(3) 若 k>1,则输出 ans 时,k^{len} 一定大于 n。(　　)

单选题

(4) 若输入的 n 等于 10^{15},输入的 k 为 1,则输出等于(　　)。

A. $(10^{30}-10^{15})/2$　　B. $(10^{30}+10^{15})/2$　　C. 1　　D. 10^{15}

(5) 若输入的 n 等于 205 891 132 094 649(即 3^{30}),输入的 k 为 3,则输出等于(　　)。

A. $(3^{30}-1)/2$　　B. 3^{30}　　C. $3^{30}-1$　　D. $(3^{30}+1)/2$

(6) 若输入的 n 等于 100 010 002 000 090,输入的 k 为 10,则输出等于(　　)。

A. 11 112 222 444 543　　B. 11 122 222 444 453

C. 11 122 222 444 543　　D. 11 112 222 444 453

3.

```
#include "algorithm"
#include "iostream"
using namespace std;

int n;
int d[50][2];
int ans;

void dfs(int n, int sum){
    if (n == 1){
        ans = max(sum, ans);
        return;
    }
    for (int i = 1;i < n;++i){
        int a = d[i - 1][0],b = d[i - 1][1];
        int x = d[i][0], y = d[i][1];
        d[i - 1][0] = a + x;
        d[i - 1][1] = b + y
        for (int j = i;j < n - 1;++j)
            d[j][0] = d[j + 1][0], d[j][1] = d[j + 1][1];
            int s = a + x + abs(b - y);
            dfs(n - 1,sum + s);
            for (int j = n - 1;j > i; -- j)
                d[j][0] = d[j - 1][0], d[j][1] = d[j - 1][1];
            d[i - 1][0] = a,d[i - 1][1] = b;
            d[i][0] = x,d[i][1] = y;
     }
}

int main(){
        cin >> n;
        for (int i = 0;i < n;++i)
        cin >> d[i][0];
        for (int i = 0;i < n;++i)
            cin >> d[i][1];
        ans = 0;
        dfs(n, 0);
        cout << ans << endl;
        return 0;
}
```

假设输入的 n 是不超过 50 的正整数，d[i][0]、d[i][1]都是不超过 10000 的正整数，完成下面的判断题和单选题。

判断题

(1) 若输入 n 为 0，此程序可能会死循环或发生运行错误。(　　)

(2) 若输入 n 为 20，接下来的输入全为 0，则输出为 0。(　　)

(3) 输出的数一定不小于输入的 d[i][0]和 d[i][1]中的任意一个。(　　)

单选题

(4) 若输入的 n 为 20，接下来的输入是 20 个 9 和 20 个 0，则输出为(　　)。

A. 1917　　B. 1908　　C. 1881　　D. 1890

(5) 若输入的 n 为 30,接下来的输入是 30 个 0 和 30 个 5,则输出为(　　)。

A. 2020　　B. 2030　　C. 2010　　D. 2000

(6) 若输入的 n 为 15,接下来的输入是 15 到 1,以及 15 到 1,则输出为(　　)。

A. 2420　　B. 2220　　C. 2440　　D. 2240

三、完善程序(单选题,每小题 3 分,共计 30 分)

1. (质因数分解)给出正整数 n,请输出将 n 质因数分解的结果,结果按从小到大输出。

例如,输入 n=120,程序应该输出 2 2 2 3 5,表示 120=2×2×2×3×5。输入保证 2≤n≤10^9。

提示:先从小到大枚举变量 i,然后用 i 不停试除 n 来寻找所有的质因子。

试补全程序。

```
01 #include "cstdio"
02 using namespace std;
03
04 int n,i;
05
06 int main(){
07     scanf("%d",&n);
08     for (i = __(1)__; __(2)__ <= n;i++){
09         __(3)__ {
10             printf("%d ",i);
11             n = n/i;
12             }
13         }
14     if (__(4)__)
15         printf("%d ", __(5)__);
16     return 0;
17 }
```

(1) 处应填(　　)。

A. n−1　　B. 0　　C. 1　　D. 2

(2) 处应填(　　)。

A. n/i　　B. n /(i*i)　　C. i*i*i　　D. i*i

(3) 处应填(　　)。

A. if (i*i<=n)　　B. if (n%i==0)

C. while (i*i<=n)　　D. while (n%i==0)

(4) 处应填(　　)。

A. n>1　　B. n<= 1　　C. i+i<=n　　D. i<n/i

(5) 处应填(　　)。

A. 2　　B. i　　C. n/i　　D. n

2. (最小区间覆盖)给出 n 个区间,第 i 个区间的左右端点是[a_i,b_i]。现在要在这些区间中选出若干个,使得区间[0, m]被所选区间的并覆盖(即每一个 0≤i≤m 都在某个所选的区间中)。保证答案存在,求所选区间个数的最小值。

输入第一行包含两个整数 n 和 m($1 \leqslant n \leqslant 5000, 1 \leqslant m \leqslant 10^9$)。

接下来 n 行,每行两个整数 a_i, b_i($0 \leqslant a_i, b_i \leqslant m$)。

提示:使用贪心法解决这个问题。先用 $O(n^2)$的时间复杂度排序,然后贪心选择这些区间。

试补全程序。

```
01 #include "iostream"     //本题需要非常强大的思维能力,难度较高。但是它是贪心问题中
02                         //非常经典的一类问题:线段覆盖问题
03 using namespace std;.
04
05 const int MAXN = 5000;
06 int n, m;
07 struct segment{int a, b;} A[MAXN];
08
09 void sort()                                  //排序
10{
11     for (int i = 0;i < n;i++)
12        for (int j = 1;j < n;j++)
13            if (   (1)   )        //两重循环,且(2)提示是一个交换的操作,因此是冒泡排序
14          {
15                segment t = A[j];
16               (2)
17            }
18 }
19
20 int main()
21{
22        cin >> n >> m;
23        for (int i = 0;i < n;i++){
24            cin >> A[i].a >> A[i].b;}
25        sort();
26        int p = 1;                           //第一次筛选
27        for (int i = 1;i < n;i++)
28            if (   (3)   ){
29                A[p++] = A[i];}
30            n = p;
31            int ans = 0,r = 0;               //第二次筛选
32            int    q = 0.
33            while (r < m)
34            {
35                while (   (4)   )
36                    q++;
37                (5) ;
38                ans++;
39             }
40            cout << ans << endl;
41            return 0;
42 }
```

(1) 处应填(　　)。

A. A[j].b < A[j−1].b　　B. A[j].b > A[j−1].b

C. A[j].a < A[j−1].a　　D. A[j].a > A[j−1].a

(2) 处应填(　　)。

A. A[j−1]=A[j];A[j]=t;　　B. A[j+1]=A[j];A[j]=t;

C. A[j]=A[j−1];A[j−1]=t;　　D. A[j]=A[j+1];A[j+1]=t;

(3) 处应填(　　)。

A. A[i].b<A[p−1].b　　B. A[i].b>A[i−1].b

C. A[i].b>A[p−1].b　　D. A[i].b<A[i−1].b

(4) 处应填(　　)。

A. q+1<n && A[q+1].b<=r　　B. q+1<n && A[q+1].a<=r

C. q<n && A[q].a<=r　　D. q<n && A[q].b<=r

(5) 处应填(　　)。

A. r=max(r, A[q+1].a)　　B. r=max(r, A[q].b)

C. r=max(r, A[q+1].b)　　D. q++

1.1.2　2020 CCF 入门级(CSP-J)参考答案

一、单项选择题(共 15 题,每题 2 分,共计 30 分)

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

二、阅读程序(除特殊说明外,判断题每题 1.5 分,单选题每题 3 分,共计 40 分)

	判断题(填√或×)				单选题	
第 1 题	(1)	(2)	(3)	(4)	(5)	(6)
	√	×	√	×	C	D
	判断题(填√或×)			单选题		
第 2 题	(1)	(2)	(3)	(4)	(5)	(6)
	×	×	√	D	A	D
	判断题(填√或×)			单选题		
第 3 题	(1)	(2)	(3)	(4)	(5)	(6)(4 分)
	×	√	×	C	B	D

三、完善程序(单选题,每小题 3 分,共计 30 分)

第 1 题					第 2 题				
(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
D	D	D	A	D	C	C	C	B	B

1.1.3　2020 CCF 入门级(CSP-J)部分试题分析

一、单项选择题分析

1. 内存储器中的每个存储单元都需用地址区分不同的存储位置。

2. 计算机能执行的唯一语言就是机器语言,即机器指令代码。

3. 选项 A、B、D 表达式均与 z 合取,故只需验证选项 C(C 为真)的结果。

4. 2048×1024×32÷8÷1024=8192KB=8MB

5. 最少的比较次数就是刚好成不下降序，比较 n－1 次；最多的比较次数是 n×(n－1)/2。这里需要理解冒泡排序与该伪码的差异。常见的冒泡排序使用标记变量一般取值 0 或 1，0 表示本趟比较没交换元素，1 反之。本例伪码标记的是一趟比较交换的最后元素的位置。

6. 3～5 行表示函数返回的是较小值，XYZ 为递归函数，故选 B。

10. 捆绑法：两个双胞胎相邻，所以方案就是 4 的全排列 4!＝24，然后双胞胎自己的全排列是 2!＝2，得数是 24×2＝48。

12. 完全二叉树的性质：满二叉树结点数是 2^h-1，h 为高度，计算一下即可知。$[\log_2 61]=6$。

14. ［方法一］ 10 个人站成一排，每班至少要 1 名，就有 9 个空，然后插入 6 个板子把他们隔开，即从 9 个里选 6 个，就是 $C_9^6=84$。

［方法二］ 每班先给一个名额，问题就变成了 7 个班级分 3 个名额。3 个名额：可以分成 3 份，两份或者一份。总方案数就是：$C_7^3+C_7^2+C_7^1=35+42+7=84$。

15. ［方法一］ 本题还是高中组合数学中的一种题型。先从 5 双手套中取完整的两双，方案是组合数 $C_5^2=10$；然后从剩下的 3 双中，取不同色的两只手套是先从 3 双中取两双 $C_3^2=3$；然后再从两双中各取一，是 $C_2^1\times C_2^1=4$；最后应用乘法原理，得数是 10×3×4＝120。

［方法二］ 先选出两副成套的，然后剩余 6 只选 1 只，然后剩下的不能匹配成套的 4 只手套选 1 只，这里会出现后面两只手套先选后选导致的重复，所以方案数要除以 2。总方案数就是：$(C_5^2\times C_6^1\times C_4^1)/2=120$。

二、阅读程序

1. 第 5 行考核字符数组初始化的概念。第 12、13 行循环测试 encoder 数组中的元素个数 k，k 值标识该数组的下一个可用单元。第 14～25 行的主要功能就是设置 encoder 数组的初值，第 26、27 行初始化数组 decoder。

encoder[]＝{C S P A B D E F G H I J K L M N O Q R T U V W X Y Z}，共 26 个字母。

decoder[]＝{D E A F G H I J K L M N O P Q C R S B T U V W X Y Z}。

字母表＝{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}。

第 28～30 行，输入字符串 st，并依据第 30 行的语句替换 st 串，显然，decoder 数组中字母序号若与字母表一致，则输出结果不变。如输入字符串 XYZ，则输出也是 XYZ。

本例程序有点类似于加密程序，关键在于看懂程序各部分的功能。各问题均可用模拟方法解决。

2. 由于数据规模很大，要正确回答题目的问题首先要看懂题目的功能。本题的功能有点像日历表或钟表，以钟表为例：每过 60 秒分针加 1，每过 60 分钟时针加 1，每过 24 小时日期加 1……只是本题的进位制与日历时钟不一样，通过输入的 k，低位由 0～k－1 循环一遍，高位加 1，每位的进制都是 k。

欲知程序的功能，最常见的方法是先模拟执行一下。假设输入 n＝10，k＝3，则初态为：n＝10，k＝3，d[0]＝0，len＝1，ans＝0；程序核心就在于第 13 行开始的双重循环内。

当 i＝0，1 时：d[0]＝1，2

当 i＝2 时：d[0]＝3

第 15 行的循环因为 len＝1 仍不执行，但 d[len－1]＝k 成立，故此时有：

d[0]＝0，d[1]＝1，len＝2，ans＝1；

当 i=5 时：j=0 时 j+1<len 成立，故此时有：

d[0]=0，d[1]=2，len=2，ans=2；

当 i=8 时：j=0 时 j+1<len 成立，故此时有：

d[0]=0，d[1]=3，len=2，ans=3；同时有：

d[len−1]=k，即 d[1]=3 成立，故此时有：

d[0]=1，d[1]=0，d[2]=1，len=3，ans=4；

程序输出 4。

分析数组 d 的变化情况就能清楚理解程序的功能。再来看看程序输出的 ans，分析代码不难发现，ans 的值实际上记录的是触发进位的次数，在第 13～28 行的代码中，核心的是两个 if 语句，每个 if 语句都是对触发进位而修正 d[]、ans 及 len 的值。len 记录的是 d 数组中有效数据个数，即 d 的实际长度。所以程序的功能是：对于 k>1 的输入，实现十进制数 n 转换为 k 进制数的计数法求解过程。

(1) 用 n=5，k=1 简单模拟一下即可知，每次都会触发进位，即 ans=n，但 len=2。k=1 时，这是本程序的一个特例，len 值不随 n 增大而增大，原因在于第 2 个 if 语句只会被执行一次。

(2) 用 n=2，k=2 简单模拟一下即可知，ans=2，len=2。

(3) 由前面的模拟可知，输入 n、k，且 k>1 时，外循环 i 从 0 到 n−1 变化，i 每 k 次递增都会触发 d[0]数据重置，即触发进位计数，而 d[1]每次数据变化都会触发计数，而 d[1]=k 时又会触发高位计数，同理可推知 d[2]，d[3]，…，d[len]的变化情况。所以有：

$$ans \leqslant k^0+k^1+\cdots+k^{len-1}=(k^{len}-1)/(k-1)$$

等比数列求和，当 n 是 k 的幂次时取最大值。

$k^{len}>n$ 成立。

(5) 因为 n=3，k=3 时 ans=1(k^0)；n=9，k=3 时 ans=4(k^0+k^1)；n=27，k=3 时 ans=13($k^0+k^1+k^2$)……求解结果符合问题(3)的分析。故本题选 A。

(6) 根据前面的分析过程可知 ans 是如何求解的，但限于问题规模太大，且 n 非 k 的整数幂，用模拟法求解该结果已不现实。

[方法一] 利用前两题的分析过程及公式直接求解。

[方法二] 本题可以利用倍数关系来求解，即每次除 k(10)，而后把每次的商相加。

方法一：利用公式求解	方法二：n/k
如 n=1000 时会产生多少次进位呢？从百位到千位，只有 1 次；从十位到百位，有 10 次；从个位到十位，有 100 次，共计 111 次。即 ans=$k^0+k^1+k^2=10^0+10^1+10^2=111$。所以对非零项求解相加即可。	1 0 0 0 1 0 0 0 2 0 0 0 0 9 1 0 0 0 1 0 0 0 2 0 0 0 0 1 0 0 0 1 0 0 0 2 0 0 0 1 0 0 0 1 0 0 0 2 0 0 1 0 0 0 1 0 0 0 2 0 1 0 0 0 1 0 0 0 2 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 1 0 0 1 0 +)1
9 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 +) 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 4 4 4 4 5 3	1 1 1 1 2 2 2 2 4 4 4 4 5 3

［方法三］　利用等比数列计算公式及程序的求解过程可以直接估算：

ans＝int((n－(k－1))/(k－1))＝int(100010002000081/9)＝11112222444453

3．为便于分析理解，先对程序代码进行分析并加注部分注释如下。

```
09  void dfs(int n, int sum){          //递归共进行 n－1 层
10    if (n == 1){
11        ans = max(sum, ans);
12        return;
13    }
14    for (int i = 1;i < n;++i){       //每层，选择数组 d 的第 0～n－1 行里相邻两行进行合并
15        int a = d[i－1][0],b = d[i－1][1];
16        int x =  d[i][0], y =  d[i][1];
17        d[i－1][0] = a +  x;
18        d[i－1][1] = b +  y
19        for (int j = i;j < n－1;++j)       //合并后把空出的地方用后面的行补上
20            d[j][0] = d[j＋1][0], d[j][1] = d[j＋1][1];
21            int s =  a＋x＋ abs(b－y);    /＊每次 sum 的增加：相邻两行第 0 列之和加第 1 列
之差的绝对值＊/
22            dfs(n－1,sum＋s);           //递归
23            for (int j = n－1;j > i; －－ j) //递归返回时还原现场操作
24                d[j][0] = d[j－1][0], d[j][1] = d[j－1][1];     //后移
25            d[i－1][0] = a,d[i－1][1] = b;     /＊还原的目的是用原始数据进行下一组相邻
行的尝试＊/
26            d[i][0] = x,d[i][1] = y;
27     }
28 }
29
30 int main(){
37         dfs(n, 0);                        //最终的 ans 是所有合并方案中 sum 值最大的
38        cout << ans << endl;
39 }
```

代码大意：函数名说明是一个深搜的算法，代码基本是按照深搜模板来的。看搜的是什么：

```
s = a + x + abs(b－y)
ans += s
```

a、b 为上一行两列数字的值，x、y 为本行两列数字的值，这里输入的值都为正值，所以输出的 ans 的最大值实际就是从第二行到最后一行 s 累加。

根据题目描述及加上的注释可知，本题是针对一个两列的二维数组，不断选出相邻的两行进行合并，直到合并成一行为止。结果为每次合并时相邻两行第 0 列之和＋第 1 列之差的绝对值。

每次选两行会产生很多种方案，ans 为其中加和最大的方案。本题与信奥经典型题“石子合并”有异曲同工之处。但此题会用更多的贪心思想。

(1) 输入 n 为 0，什么都没做，因为第 10 和 14 行都不成立，程序会立即返回主函数，直接输出 ans＝0，结束程序，不会发生错误。

(2) 输入 n 为 20,接下来的输入全为 0,运算过程中所有 s 都是 0,ans 也是 0。

(3) 这里减法,所以 ans 可能小于输入的 d[i][1],例如,若 n=1,ans=0,是小于 d 数组的。

输入:

2 0 0 2 3

输出:

1

(4) 第二列为 0,可以忽略,用 Sn 表示前 n 项的和,这里计算的就是:

S2+S3+S4+…+S20=2×9+3×9+4×9+…+20×9=1881。

(5) 这里不会出现 b<y 的情况,所以可以去掉绝对值符号,用 Sn 表示前 n 项的和,这里计算的就是:

S1-5+S2-5+S3-5+…+S29-5=0×5+1×5+2×5+…+28×5=2030。

(6) 因为 n 较大,难以发现规律,可以让 n=4 或 n=6 为例,模拟执行并找出规律。

输入的两列数字是一样的,计算的内容:

s=a+x+abs(b-y); ans+=s;

此时不会出现 b<y 的情况,所以可以去掉绝对值符号,x,y 抵消,s=a+b,

所以 ans=2×(S1+S2+S3+S4+…+S14)=2240。

本题的思维分析及计算工作量非常大,考场上需要根据自己的特点采取一些合适的策略,只有找出规律才能确保正确选择。

三、完善程序

1. 质因数分解

(1) 因子最小为 2,所以选 i=2。

(2) 因子最大为$\sqrt{n}$,所以选 i*i。

(3) 由题目可知,一个因子可能被分解出好多次,所以选 D。

(4) 分解完成后,n 的值为 1 或者质数,否则还有>$\sqrt{n}$的质因数。

(5) 不是 1 的话需要单独输出,即剩下的为质数。

2. 最小区间覆盖

本题需要非常强大的思维能力,难度较高。问题本质上其实是贪心问题中非常经典的一类问题:线段覆盖问题。sort()函数由两重循环构成,且(2)提示是一个交换的操作,因此是冒泡排序,主函数中做了两次筛选操作。

(1) 按照区间起点进行排序,选 C。

(2) 基础的交换代码,选 C。

(3) 筛掉起点靠后同时终点靠前的区间,这样的区间不可能选用,选 C。

(4) 此时剩余的区间逐个选用,选 B。

(5) 选用最后一个区间,更新 r,选 B。

说明:本题是经典的线段覆盖问题,所以不熟悉的读者需要自己补习。

1.2　2019 入门级(CSP-J)C++语言试题 A 卷及解析

1.2.1　2019 CCF 入门级(CSP-J)试题

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

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(　　)。

A. 01 0010 1000 1011　　B. 01 0010 1001 0011

C. 01 0010 1000 0001　　D. 01 0010 1000 0011

3. 一个 32 位整型变量占用(　　)字节。

A. 32　　B. 128　　C. 4　　D. 8

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>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;

5. 有 100 个已经排好序的数据元素,采用折半查找时,最大的比较次数为(　　)。

A. 7　　B. 10　　C. 6　　D. 8

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. 所需要的空间与线性表长度成正比　　D. 可随机访问任何一个元素

7. 把 8 个同样的球放在 5 个同样的袋子里,允许有空袋子,总共有多少种分法?(　　)

A. 22　　B. 24　　C. 18　　D. 20

8. 一棵二叉树如图 1-1 所示,若采用顺序存储结构,即用一维数组元素存储该二叉树中的结点(根结点的下标为 1,若某结点的下标为 i,则其左孩子位于下标 2i 处、右孩子位于下标 2i+1 处),则该数组的最大下标至少为(　　)。

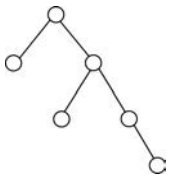

图 1-1

A. 6　　B. 10　　C. 15　　D. 12

9. 100 以内最大的素数是(　　)。

A. 89　　B. 97　　C. 91　　D. 93

10. 319 与 377 的最大公约数是(　　)。

A. 27　　B. 33　　C. 29　　D. 31

11. 小胖想减肥,教练给他制定了两套方案。方案一:每次连续跑 3km 可以消耗 300 千卡(耗时半小时);方案二:每次连续跑 5km 可以消耗 600 千卡(耗时 1 小时)。

小胖每周周一到周四可以抽出半小时跑步,周五到周日能抽出 1 小时跑步。另外,教练建议小胖每周最多跑 21km,否则会损伤膝盖。请问小胖每周最多通过跑步消耗多少千卡?(　　)

A. 3000　　B. 2500　　C. 2400　　D. 2520

12. 一副纸牌中除掉大小王有 52 张牌,4 种花色,每种花色 13 张。假设从这 52 张牌中随机选择 13 张,则至少(　　)张牌花色一样。

A. 4　　B. 2　　C. 3　　D. 5

13. 一些数字可以颠倒过来看,例如 0、1、8 颠倒过来还是其本身,6 颠倒过来是 9,9 颠倒过来是 6,其他数字颠倒过来则构不成数字。类似地,一些多位数也可以颠倒过来看,例如 106 颠倒过来是 901。假如某个城市的车牌是由 5 位数字组成,每一位可以取 0～9,问这个城市最多几个车牌颠倒过来恰好是其本身?(　　)

A. 60　　B. 125　　C. 75　　D. 100

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,则其后序遍历序列为(　　)。

A. ABCDEFGHIJ　　B. ABDEGHJCFI

C. ABDEGJHCFI　　D. ABDEGHJFIC

15. 以下哪个奖项是计算机科学科学领域的最高奖项?(　　)

A. 图灵奖　　B. 鲁班奖　　C. 诺贝尔奖　　D. 普利策奖

二、阅读程序(程序输入不超过数组或字符串定义的范围;判断题正确填√,错误填×;除特殊说明外,判断题每题 1.5 分,单选题每题 3 分,共计 40 分)

1.

```
01  #include<cstdio>
02  #include<cstring>
03  using namespace std;
04  char st[100];
05  int main(){
06      scanf("%s", st);
07      int n=strlen(st);
08      for (int i=1;i<=n;++i){
09      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("%s", st);
16      return 0;
17  }
```

判断题

(1) 输入的字符串只能由小写字母或大写字母组成。(　　)

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

(3) 若将第 8 行的“i<=n”改为“i*i<=n”,程序运行结果不变。(　　)

(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同输入的字符串一样。(　　)

单选题

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

A. 18　　B. 6　　C. 10　　D. 1

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

A. 36　　B. 100000　　C. 1　　D. 128

2.

```
01  #include<cstdio>
02  using namespace std;
03  int n, m;
04  int a[100], b[100];
05
06  int main(){
07      scanf("%d%d", &n, &m);
08      for (int i=1;i<=n;++i){
09          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", ans);
30      return 0;
31  }
```

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

判断题

(1) 当m>0时,输出的值一定小于2n。(　　)

(2) 执行完第27行的"++ans"时,ans一定是偶数。(　　)

(3) a[i]和 b[i]不可能同时大于 0。(　　)

(4) 若程序执行到第 13 行时,x 总是小于 y,那么第 15 行不会被执行。(　　)

单选题

(5) 若 m 个 x 两两不同,且 m 个 y 两两不同,则输出的值为(　　)。

A. 2n－2m　　B. 2n＋2　　C. 2n－2　　D. 2n

(6) 若 m 个 x 两两不同,且 m 个 y 都相等,则输出的值为(　　)。

A. 2n－2　　B. 2n　　C. 2m　　D. 2n－2m

3.

```
01  #include<iostream>
02  using namespace std;
03  const int maxn=10000;
04  int n;
05  int a[maxn];
06  int b[maxn];
07  int f(int l, int r, int depth){
08          if (l>r){
09              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  }
```

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。(　　)

(2) 如果 b 数组全为 0,则输出为 0。(　　)

单选题

(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

(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

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

A. 386　　B. 383　　C. 384　　D. 385

(6) 当 n=100 时,若 b 数组满足,对任意 0<i<71,都有 b[i]=1,那么输出最小为(　　)。

A. 582　　B. 580　　C. 579　　D. 581

三、完善程序(单选题,每小题 3 分,共计 30 分)

1. (**矩阵变幻**)有一个奇幻的矩阵,在不停地变幻,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 $\begin{bmatrix}1&1\\1&0\end{bmatrix}$,最初该矩阵只有一个元素 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 次后为 $\begin{bmatrix}0&0\\0&1\end{bmatrix}$,矩阵变幻 2 次后为 $\begin{bmatrix}0&0&0&0\\0&1&0&1\\0&0&1&1\\0&1&1&0\end{bmatrix}$。

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,$(11)_2 << 2 = (1100)_2$

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。

```
01  #include <cstdio>
02  using namespace std;
03  int n;
04  const int max_size = 1 << 10;
05
06  int res[max_size][max_size];
07
08  void recursive(int x, int y, int n, int t){
09      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  }
```

(1) ①处应填(　　)。

A. n%2　　B. 0　　C. t　　D. 1

(2) ②处应填(　　)。

A. x−step,y−step　　B. x,y−step

C. x−step,y　　D. x,y

(3) ③处应填(　　)。

A. x−step,y−step　　B. x+step,y+step

C. x−step,y　　D. x,y−step

(4) ④处应填(　　)。

A. n−1,n%2　　B. n,0　　C. n,n%2　　D. n−1,0

(5) ⑤处应填(　　)。

A. 1<<(n+1)　　B. 1<<n　　C. n+1　　D. 1<<(n−1)

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 3 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:$1<n<10^7$,$1<a[i],b[i]<10^4$。

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。

```
#include <cstdio>
#include <cstring>
using namespace std;
const int maxn = 10000000;
const int maxs = 10000;

int n;
unsigned a[maxn], b[maxn],res[maxn], ord[maxn];
unsigned cnt[maxs + 1];

int main(){
    scanf("%d", &n);
    for (int i = 0;i < n;++i){
        scanf("%d%d", &a[i], &b[i]);}
    memset(cnt, 0, sizeof(cnt));
    for (int i = 0;i < maxs;++i){
        ____①____; }                          //利用 cnt 数组统计数量
    for (int i = 0;i < n;++i){
        cnt[i + 1] += cnt[i];}
    for (int i = 0;i < n;++i)
        ____②____;                            //记录初步排序结果
    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", ____⑤____ );}
31      return 0;
32  }
```

(1) ①处应填(　　)。

A. ++cnt [i]　　B. ++cnt[b[i]]

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

(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

(3) ③处应填(　　)。

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

C. ++cnt[a[i]]　　D. ++cnt [i]

(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]

(5) ⑤处应填(　　)。

A. a[i], b[i]　　B. a[res[i]], b[res[i]]

C. a[ord[res[i]]]j b[ord[res[i]]]　　D. a[res[ord[i]]]j b[res[ord[i]]]

1.2.2　2019 CCF 入门级(CSP-J)参考答案

一、单项选择题(共 15 题,每题 2 分,共计 30 分)

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

二、阅读程序(除特殊说明外,判断题每题 1.5 分,单选题每题 3 分,共计 40 分)

第 1 题	判断题(填√或×)				单选题	
	(1)	(2)	(3)	(4)	(5)	(6)
	×	√	×	√	B	B
第 2 题	判断题(填√或×)				单选题	
	(1)	(2)	(3)	(4)	(5)	(6)
	√	×	×	×	A	A
第 3 题	判断题(填√或×)		单选题			
	(1)	(2)	(3)	(4)	(5)	(6)
	×	√	A	D	D	B

三、完善程序(单选题,每小题 3 分,共计 30 分)

第 1 题					第 2 题				
(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
C	D	B	B	B	B	D	C	A	B

1.2.3 2019 CCF 入门级(CSP-J)部分试题分析

一、单项选择题分析

1. 信息网络常识,中国的顶级域名是.cn。

2. 逻辑位运算基本知识,当且仅当两位上均为 1 的时候结果才为 1。

0&0=0; 1&0=0; 0&1=0; 1&1=1

3. 常识,一字节是 8 位,32/8=4。

4. s 初始化为 a,for 循环执行了 c 次-1,所以 s=a-c。

5. 即二分法,每次比较可以缩减一半的范围,$2^6 < 100 < 2^7$,所以查找 7 次。

6. 链表只能从有标记的头尾指针依次访问元素,无法随机访问。

7. 因为球和袋子一样,即求把 8 拆分成 1~5 个数有多少种方法。按个数枚举得 1 个数到 5 个数分别有 1、4、5、5、3 种,总共 18 种。

【详细分析】保证分法升序就能没有遗漏枚举了,实际考场中采用此方法是最稳妥的,可以避免推错。

问题:把 8 个同样的球放在同样的 5 个袋子里,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,无论是放哪个袋子,都只算同一种分法。

解析:把问题合成,先思索 5 个袋子都不空的状况,再思索 4 个袋子不空的状况,以此类推,最后思索只运用一个袋子的状况(这种分法只要 1 种),把一切子状况的分法数相加求出总分法。

进一步剖析,运用 k 个袋子装 n 个球(袋子不空),一共有几种分法的问题能够转换为 k 个数相加等于 n 的种数问题。

运用 5 个袋子分 8 个球则有 3 种: 1+1+1+1+4=8 1+1+1+2+3=8 1+1+2+2+2=8 运用 4 个袋子分 8 个球则有 5 种: 1+1+1+5=8 1+1+2+4=8 1+1+3+3=8 1+2+2+3=8 2+2+2+2=8	运用 3 个袋子分 8 个球则有 5 种: 1+1+6=8 1+2+5=8 1+3+4=8 2+2+4=8 2+3+3=8 运用 2 个袋子分 8 个球则有 4 种: 1+7=8 2+6=8 3+5=8 4+4=8 运用 1 个袋子分 8 个球则有 1 种: 8=8

因而,该问题的答案即为一切子状况下的和,3+5+5+4+1=18。