

# 2020CCF 非专业级别软件能力认证第一轮 X 卷

## (CSP-S) 提高级 C++语言试题卷

认证时间: 2020 年 10 月 17 日 14:30-16:30

考生注意事项:

- 试题纸共有 12 页, 答题纸共有 1 页, 满分 100 分。请在答题纸上作答, 写在试题纸上的一律无效。
- 不得使用任何电子设备 (如计算器、手机、电子词典等)或查阅任何书籍资料。

一、选择题 (以下共有 15 道题目, 对于每道题目, 在 ABCD 选项中选择正确的一项。每题 2 分, 共 30 分)

1. 以下关于 CCF 组织举办的竞赛说法不正确的是

- A. CSP 认证每年举办一次, 分“入门组”和“提高组”, 分两轮进行。
- B. NOIP 竞赛有悠久的历史, 最早于 1996 年开始举办。
- C. NOI 竞赛始于 1983 年, 该竞赛不仅可以现场参加, 还可以通过网上报名参加同步赛。
- D. 各省的省队选拔独立进行, 分两轮, 初中选手只拥有 E 类名额。

2. 以下关于个人计算机操作系统的说法正确的是

- A. Bill Gates, 他创办的 Microsoft 公司开发了 Linux 系列系统。
- B. Steve Jobs, 他的苹果公司开发了 IOS 系统。
- C. 目前 Linux 和 Windows 系统都是开源的, 网上都能搜索并下载。
- D. 除了以上提到的所有操作系统, 还有 DOS, Unix 等系统。Windows 使用最广

3. 十进制表达式  $(3 \times 512 + 7 \times 64 + 4 \times 8 + 3) + 1$  的结果以二进制形式表示为

- A. 10111100101 B. 11111100100 C. 111110100011 D. 11111101101

4. 计算机能直接执行的指令包括两部分, 他们是

- A. 源操作数与目标操作数 B. 操作码与操作数
- C. ASCII 码与汉字代码 D. 数字与字符

5. 以下程序段:

```
for (i=1; i<=n; i++) a[i]=i;
for (i=2; i<=n; i++)
    if (i==a[i])
    {
        for (j=1; j*i<=n; j++)
            a[j*i]=a[j*i]/i*(i-1);
    }
```

的作用是求出 1 至 n 的所有

- A. 质数 B. 整数的最大因数 C. 整数的因数个数 D. 整数的  $\phi(i)$

6. 关于 BIOS 下面说法正确的是

- A. BIOS 一般由操作系统厂商来开发完成。
- B. BIOS 里包含了键盘、鼠标、声卡、显卡、打印机等常用输入输出设备的驱动程序。
- C. BIOS 是计算机基本输入输出系统软件的简称。

D. BIOS 能提供各种文件拷贝、复制、删除以及目录维护等文件管理功能。

7.命题“ $P \rightarrow Q$ ”可读做 P 蕴含 Q，其中 P、Q 是两个独立的命题。只有当命题 P 成立而命题 Q 不成立时，命题“ $P \rightarrow Q$ ”的值为 false，其它情况均为 true。与命题“ $P \rightarrow Q$ ”等价的逻辑关系式是。

A.  $\neg P \wedge Q$  B.  $P \wedge Q$  C.  $\neg (P \vee Q)$  D.  $\neg (\neg Q \vee P)$

8.现在有一个集合，里面包含了字符串“zlymAKIOItxdy”（不含引号）的所有字符，则请问该集合的所有非空真子集的个数为

A.4095 B.4096 C.4094 D.2046

9.对于完全背包问题(给出 n 种物品和一个容积为 m 的背包，每种物品有无限个，单个大小为  $v[i]$ ，价值为  $w[i]$ ，要求选择适当的物品放入背包，满足大小不超过容积且价值最大)，设  $f[i]$  表示用去 i 的空间能获得的最大价值，倒序枚举 i 为使用的空间，正序枚举 j 为物品编号，则可写出动态转移方程

A. $f[i]=\max(f[i],f[i-w[j]]+v[j])$

B. $f[i]=\max(f[i],f[i-v[j]]+w[j])$

C. $f[i]=\min(f[i],f[i-w[j]]+v[j])$

D. $f[i]=\min(f[i],f[i-v[j]]+w[j])$

10.小明要去南美洲旅游，一共乘坐三趟航班才能到达目的地，其中第 1 个航班准点的概率是 0.9，第 2 个航班准点的概率为 0.8，第 3 个航班准点的概率为 0.9。如果存在第 i 个 ( $i=1,2$ ) 航班晚点，第  $i+1$  个航班准点，则小明将赶不上第  $i+1$  个航班，旅行失败；除了这种情况，其他情况下旅行都能成功。请问小明此次旅行成功的概率是

A. 0.5 B. 0.648 C. 0.72 D. 0.74

11.以下关于 CSP 非专业级认证与 NOIP 竞赛关系的说法最恰当的是

A.完全无关 B.组织者相同 C.举办目标相同 D.从属关系，CSP 从属于 NOIP

12.关于信息学竞赛涉及算法，以下说法不正确的是

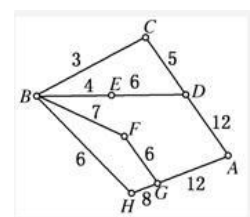
A.搜索分为深度优先和广度优先两种，特点是代码难度低，但都很费时

B.模拟是一种最基本的算法，就是按照题目要求做。广义的模拟还包含其他算法

C.贪心算法类似于“鼠目寸光”，该算法前提是满足“部分最优组成全局最优”

D.动态规划包含了很多类型和优化，动态转移方程是该算法的核心

13.如图，小圆圈表示网络的结点，结点之间的连线表示它们有网线相连，连线标注的数字表示该段网线单位时间内可以通过的最大信息量，现从结点 B 向结点 A 传递信息，信息可以分开沿不同的路线同时传递，则单位时间内传递的最大信息量为



A、19

B、20

C、24

D、25

14.分辨率为 1600x900、16 位色的位图，存储图像信息所需的空间为

A. 2812.5KB B. 4218.75KB C. 4320KB D. 2880KB

15.情境：一位妈妈生了 n 胞胎，这 n 个孩子长得非常相似，让人无法辨认。一天晚上，妈妈要给这 n 个孩子洗澡。妈妈每次从孩子中抓出一个洗澡，洗完后又把他放回孩子之中，如此重复 n 次。问题：若  $n=10$ ，则在所有的可能中，恰好只有三个孩子没有洗澡的可能性约为

A.14% B.36% C.31% D.17%

二、阅读程序（以下共有三段程序。每段程序后面有 6 个题目，前 4 题为判断题，后 2 题为选择题。对于判断题，选择 A 代表正确，B 代表错误；对于选择题，请在 ABCD 选项中选择正确的一项。判断题每题 1 分，选择题每题 3 分，共 30 分）

①

```
#include<cstdio>
#define MAX_N 20
#define ll long long
using namespace std;
int n;
long long f[MAX_N][MAX_N];
int main()
{
    scanf("%d",&n);
    for(int i=0;i<=n;i++)
    {
        f[0][i]=1;
    }
    for(int i=1;i<=n;i++)
    {
        for(int j=i;j<=n;j++)
        {
            if(i==j)f[i][j]=f[i-1][j];
            else f[i][j]=f[i][j-1]+f[i-1][j];
        }
    }
    Cout<<f[n][n]<<endl;
    return 0;
}
```

16.如果第 3 行的代码缺失，程序会出现编译错误。

17.本程序中 f 数组内被改变过的元素的值与 n 无关，即对于不同的 n，f 数组中不为 0 的元素值都对应相同

18.若  $n=10$ ，对于任何  $0 \leq i \leq n$ ， $f[0][i]=1$

19.若  $n=10$ ，对于任何  $0 \leq i \leq n$ ， $f[1][i]=1$

20.本程序没有使用的基本算法是

A. 模拟                      B.递推                      C.递归                      D.动态规划

21.对于输入 6，本程序输出

A. 5                      B.14                      C.42                      D.132

②

```
#include<iostream>
#include<algorithm>
using namespace std;
long int n,k,sum[100001],num=1,f[100001];
struct ren
{

```

```

        long int ks,js;
};
ren z[100001];
int cmp(ren a,ren b)
{
    return a.ks>b.ks;
}
int main()
{
    long int i,j;
    cin>>n>>k;
    for(i=1;i<=k;i++)
    {
        cin>>z[i].ks>>z[i].js;
        sum[z[i].ks]++;
    }
    sort(z+1,z+k+1,cmp);
    for(i=n;i>=1;i--)
    {
        if(sum[i]==0)
            f[i]=f[i+1]+1;
        else for(j=1;j<=sum[i];j++)
        {
            if(f[i+z[num].js]>f[i])
                f[i]=f[i+z[num].js];
            num++;
        }
    }
    cout<<f[1]<<endl;
}

```

22.本程序使用了贪心算法

23.第 23 行 sort 排序后 z 数组中的元素按 js 元素降序排序

24.以下是对于本代码的一段输入，则对应的输出是 3

15 6

1 2

1 6

4 11

8 5

8 1

11 5

25.本代码的 24 行换成 for (i=1;i<=n;++i)，可能会引发“数组越界”错误

26.第 10 至 12 行程序定义了一个 cmp 函数以用于 sort 的比较。我们可以用含有以下的一个选项的程序段来定义适用于该结构体类型的小于运算。这个选项是

A. this

B.operator

C.iterator

D.it

27.本程序的时间复杂度是

A.O(n)

B.O(k)

C.O(n • log(n))

D.O(nk)

③

//注释：本程序使用一种树形数据结构，实现了将读入的数据按照“左子树<=根<=右子树”的方法建树，并输出树的深度及后序遍历。

```
#include<iostream>
using namespace std;
struct node
{
    int data,left,right;
};
node tree[300010];
int n1;
int ans;
int NNN(int x)
{
    n++;
    tree[n].data=x;
    tree[n].left=tree[n].right=0;
    return n;
}
void insert(int x,int& idx)
{
    if(!idx)
    {
        idx=NNN(x);
        return ;
    }
    if(x<tree[idx].data)
        insert(x,tree[idx].left);
    else
        insert(x,tree[idx].right);
}
void dfs(int idx,int deep)
{
    if(!idx)
    {
        ans=max(ans,deep);
        return ;
    }
    dfs(tree[idx].left,deep+1);
    dfs(tree[idx].right,deep+1);
}
void printhx(int idx)
```

```

{
    if(idx)
    {
        printhx(tree[idx].left);
        printhx(tree[idx].right);
        cout<<tree[idx].data<<endl;
    }
}
int main(){
    cin>>n1;
    for(int i=1;i<=n1;i++)
    {
        int x;
        cin>>x;
        int id=1;
        if(i==1)
            NNN(x);
        else
            insert(x,id);
    }
    dfs(1,0);
    cout<<"deep="<<ans<<endl;
    printhx(1);
    return 0;
}

```

28.在本程序中，第 6 行的数组必须开到数据范围的结点数的 4 倍以上

29.存放树的节点时运用了指针思想

30.程序第 9 至 14 行的 NNN 函数作用是给树增加一个空节点

31.将程序第 30 行和 31 行交换，不会对程序产生影响

32.一种树形结构是指

A. 权值线段树      B. 树状数组      C. 斐波那契堆      D. 二叉搜索树

33.以下是一个对于本程序的输入，则与之对应的输出是

8

1 4 3 9 10 35 2 7

A	B	C	D
deep=4	deep=6	deep=5	deep=5
1	4	2	2
2	2	3	3
3	7	7	7
4	1	10	35
7	9	35	10
9	10	9	9
10	3	1	4
35	35	4	1

三、补充程序（本大题共含有 2 篇代码，共 10 小题，每小题 4 分，共 40 分。请在每道小题后所给的四条代码中选出最恰当的一项，使这段代码填入完整程序中对应的空缺处能符合题意。）

① 《Qjl 的比赛》（20 分）

Qjl 是一名神犇，他最近出了一场盛大的 AK 赛。比赛共含有  $n(1 \leq n \leq 100000)$  题，由于是 AK 赛，任何人做了任何一道题都能保证 AC。但是 Qjl 身为神犇，要摆神犇的架子，具体做法是对于不同的题，赋予不同的满分，这样会让 AK 者的分数变得好看。不过，他的比赛中不断会出现出锅的情况，也就是题目满分应该被调整。他会不断调整一些题目的满分，使之更合理。现在 Qjl 需要你编写程序，帮他满足两种操作，如果你写出了代码，你有可能会获得 2147483648%32768 分的额外满分！

他首先给你两个正整数  $n, m$ ，表示题目数与操作数；然后他会给你  $n$  个正整数，表示这些题目初始的满分。接下来他会不断给你发  $m$  条操作指令：

1 x y k: 表示将题号  $i \in [x, y]$  的题目的满分都加上  $k$ ；2 x y: 表示询问题号  $i \in [x, y]$  的所有题目的 AK 分（即总分和。）请你编写程序，以享受 AK 的快乐。

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
struct point
{
    int lft,rgt,num,lzy;
} a[400401];
int b[100001],n,m,i,x,y,z,t;
void build(int p,int l,int r)
{
    a[p].lft=l;
    a[p].rgt=r;
    if (l==r)
    {
        a[p].num=b[l];
        return;
    }
    build(____①____,l,(l+r)/2);
    build(____②____,(l+r)/2+1,r);
    a[p].num=a[p*2].num+a[p*2+1].num;
}
void down(int p)
{
    if (____③____)
    {
        a[p*2].num+=a[p].lzy*(a[p*2].rgt-a[p*2].lft+1);
        a[p*2+1].num+=a[p].lzy*(a[p*2+1].rgt-a[p*2+1].lft+1);
        a[p*2].lzy+=a[p].lzy;
        a[p*2+1].lzy+=a[p].lzy;
```

```

        _____④;
    }
}
void add2(int nd,int st,int ed,int x)
{
    if (st<=a[nd].lft && ed>=a[nd].rgt)
    {
        a[nd].num+=_____⑤;
        a[nd].lzy+=x;
        return;
    }
    down(nd);
    if (st<=(a[nd].lft+a[nd].rgt)/2) add2(nd*2,st,ed,x);
    if (ed>=(a[nd].lft+a[nd].rgt)/2+1) add2(nd*2+1,st,ed,x);
    a[nd].num=a[nd*2].num+a[nd*2+1].num;
}
int Find(int nd,int l,int r)
{
    if (_____⑥) return _____⑦;
    down(nd);
    int bb=0;
    if (l<=(a[nd].lft+a[nd].rgt)/2) bb+=Find(nd*2,l,r);
    if (r>=(a[nd].lft+a[nd].rgt)/2+1) bb+=Find(nd*2+1,l,r);
    return bb;
}
signed main()
{
    scanf("%lld%lld",&n,&m);
    for (i=1;i<=n;i++)
    {
        scanf("%lld",&b[i]);
    }
    build(1,1,n);
    for (i=1;i<=m;i++)
    {
        scanf("%lld",&t);
        if (t==1)
        {
            scanf("%lld%lld%lld",&x,&y,&z);
            add2(1,x,y,z);
        }
        else
        {
            scanf("%lld%lld",&x,&y);

```



```

        printf("%lld\n",Find(1,x,y));
    }
}

```

34.填入①和②的代码分别是

- A.p\*2; p\*2+1
- B.p\*2+1; p\*2
- C.p+1; p+2
- D.p+2; p+1

35 填入③处的代码是

- A.a[p].num==0
- B.a[p].num!=0
- C.a[p].lay==0
- D.a[p].lzy!=0

36.填入④处的代码是

- A.a[p].num=0
- B.a[p].lzy=0
- C.a[p].num=a[p].lzy=0
- D.{down(p\*2);down(p\*2+1);}

37.填入⑤处的代码是

- A.a[nd].rgt-a[nd].lft+1
- B.a[nd].rgt-a[nd].lft
- C.x\*(a[nd].rgt-a[nd].lft+1)
- D.x\*(a[nd].rgt-a[nd].lft)

38.填入⑥和⑦处的代码分别是

- A.l<=a[nd].lft && a[nd].rgt<=r; a[nd].num
- B.l<=a[nd].lft && a[nd].rgt<=r; a[nd].num\*(l-r+1)
- C.a[nd].lft<=l && r<=a[nd].rgt; a[nd].num\*(l-r+1)
- D.a[nd].lft<=l && r<=a[nd].rgt; a[nd].num

## ② 《奶牛也感染》（20 分）

每头奶牛都不想成为被新型病毒感染的奶牛。被所有奶牛接触的奶牛就是一头被感染的奶牛。因为奶牛喜欢偷偷摸摸地行动，所有接触是单向的，所有奶牛都与自己接触。奶牛之间的接触是可以传递的——如果 A 接触过 B，B 接触过 C，那么 A 也接触过 C。牛栏里共有 N 头奶牛，给定一些奶牛之间的接触关系，请你算出有多少头奶牛不幸被感染。

输入的第一行是两个用空格分开的整数 N 和 M。接下来 M 行每行两个用空格分开的整数 A 和 B，表示 A 接触过 B。输出被感染的奶牛的数量。

```

#include<bits/stdc++.h>
using namespace std;
struct edge
{
    int v,next;
} p[50001];
int front[10001],M;

```

```

void Add(int u,int v)
{
    ++M;
    p[M].v=v;
    ①
    front[u]=M;
}
int dfn[10001],low[10001],N,s;
stack<int> zhan;
int which[10001];
int Out[10001],id,large[10001];
int n,m;
int dfs(int x)
{
    ++N;
    dfn[x]=low[x]=N;
    zhan.push(x);
    for (int i=front[x];i=p[i].next)
    {
        if ( ② )
        {
            dfs(p[i].v);
            ③
        }
        else
            low[x]=min(low[x],dfn[p[i].v]);
    }
    if (dfn[x]==low[x])
    {
        ++S;
        int S=0;
        int v;
        do
        {
            v=zhan.top();
            zhan.pop();
            ④
            ++S;
        } while (x!=v);
        Large[s]=S;
    }
}
int main()
{

```

```

int i,j;
scanf("%d%d",&n,&m);
for (i=1;i<=m;++i)
{
    int x,y;
    scanf("%d%d",&x,&y);
    Add(x,y);
}
for (i=1;i<=n;++i)
    if (dfn[i]==0) dfs(i);
for (i=1;i<=n;++i)
    for (j=front[i];j;p[j].next)
        if (which[p[j].v]!=which[i]) ⑤;
for (i=1;i<=s;++i)
    if (Out[i]==0)
    {
        if (id!=0)
        {
            printf("%d\n",0);
            return 0;
        }
        id=i;
    }
    printf("%d\n",large[id]);
return 0;
}

```

39.①处填入的代码是

- A.p[M].next=u
- B.p[M].next=v
- C.p[M].next=front[u]
- D.p[M].next=front[v]

40.②处填入的代码是

- A.!dfn[p[i].v]
- B.dfn[p[i].v]
- C.low[p[i].v]
- D.!low[p[i].v]

41.③处填入的代码是

- A.dfn[x]=min(dfn[x],dfn[p[i].v]);
- B.dfn[x]=max(dfn[x],low[p[i].v]);
- C.low[x]=max(low[x],dfn[p[i].v]);
- D.low[x]=min(low[x],low[p[i].v]);

42.④处填入的代码是

- A.++s;
- B.which[v]=s;

C.x=zhan.top(),zhan.pop();

D.zhan.push(x+v);

43.⑤处填入的代码是

A.++Out[which[i]]

B.++which[Out[i]

C.++Out[Out[i]]

D.++which[which[i]]

说明：试题卷至上一页结束，以下为答案和答题卷（不怎么好看，选择使用），请在下发试题之前把以下两页保存到新的文档后删除。

答案：

1	2	3	4	5	6	7	8	9	10
A	D	B	B	D	C	D	D	B	D
11	12	13	14	15	16	17	18	19	20
B	A	A	A	B	B	A	A	B	C
21	22	23	24	25	26	27	28	29	30
D	B	B	B	B	B	D	B	A	A
31	32	33	34	35	36	37	38	39	40
A	D	D	A	D	B	C	D	C	B
41	42	43							
D	B	A							

## (CSP-S) 提高级 C++语言答题卷

**考生注意事项:**

- 试题纸共有 11 页, 答题纸共有 1 页, 满分 100 分。请在答题纸上作答, 写在试题纸上的一律无效。
- 不得使用任何电子设备 (如计算器、手机、电子词典等) 或查阅任何书籍资料。
- 请将每题的答案按照题号先填入方格, 再填涂。

语言:

准考证号					
------	--	--	--	--	--

[0]	[0]	[0]	[0]	[0]
[1]	[1]	[1]	[1]	[1]
[2]	[2]	[2]	[2]	[2]
[3]	[3]	[3]	[3]	[3]
[4]	[4]	[4]	[4]	[4]
[5]	[5]	[5]	[5]	[5]
[6]	[6]	[6]	[6]	[6]
[7]	[7]	[7]	[7]	[7]
[8]	[8]	[8]	[8]	[8]
[9]	[9]	[9]	[9]	[9]

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

16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33
16	[A] [B]		22	[A] [B]		28	[A] [B]	
17	[A] [B]		23	[A] [B]		29	[A] [B]	
18	[A] [B]		24	[A] [B]		30	[A] [B]	
19	[A] [B]		25	[A] [B]		31	[A] [B]	
20	[A] [B] [C] [D]		26	[A] [B] [C] [D]		32	[A] [B] [C] [D]	
21	[A] [B] [C] [D]		27	[A] [B] [C] [D]		33	[A] [B] [C] [D]	

三、补充程序（共 10 小题，每小题 4 分，共 40 分。）

34	35	36	37	38	39	40	41	42	43
34	[A] [B] [C] [D]		37	[A] [B] [C] [D]		40	[A] [B] [C] [D]		
35	[A] [B] [C] [D]		38	[A] [B] [C] [D]		41	[A] [B] [C] [D]		
36	[A] [B] [C] [D]		39	[A] [B] [C] [D]		42	[A] [B] [C] [D]		
						43	[A] [B] [C] [D]		