

2021 LGR 非专业级别软件能力认证第一轮

(SCP) C++语言模拟试题

认证时间：2021 年 9 月 4 日 09:30~11:30

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。

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. 以补码存储的 8 位有符号整数 10110111 的十进制表示为（ ）。
A. -73 B. 183 C. 72 D. -72
2. 现有一段 24 分钟的视频文件，它的帧率是 30Hz，分辨率是 1920×1080，每帧图像都是 32 位真彩色图像，使用的视频编码算法达到了 25% 的压缩率。则这个视频文件占用的存储空间大小约是（ ）。
A. 668GiB B. 334GiB C. 85GiB D. 500GiB
3. 链接器的功能是（ ）。
A. 把源代码转换成特定硬件平台的机器指令
B. 把机器指令组合成完整的可执行程序
C. 把源代码转换成可执行程序
D. 把高级语言翻译成低级语言
4. 对一个 n 个顶点， m 条边的带正权有向简单图使用 Dijkstra 算法计算单源最短路时，如果再使用一个可以在 $\Theta(\log n)$ 时间复杂度内查询堆内最小值、在 $\Theta(\sqrt{n})$ 时间复杂度内合并两个堆、在 $\Theta(1)$ 时间复杂度内将堆内一个元素变小、在 $\Theta(\log n)$ 时间复杂度内弹出堆内最小值的堆优化 Dijkstra 算法，则整个 Dijkstra 算法的时间复杂度为（ ）。
A. $\Theta(n\sqrt{n} + m \log n)$
B. $\Theta((n + m) \log n)$
C. $\Theta(m + n \log n)$
D. $\Theta(m\sqrt{n} + n \log n)$
5. 具有 n 个顶点， m 条边的连通图采用邻接矩阵存储结构，进行深度优先遍

-
- A. 艾伦·麦席森·图灵 (Alan Mathison Turing)
 - B. 约翰·冯·诺依曼 (John von Neumann)
 - C. 克劳德·艾尔伍德·香农 (Claude Elwood Shannon)
 - D. 罗伯特·塔扬 (Robert Tarjan)

15. 在计算机非专业级别软件能力认证 CSP-S 进行时, 下列行为中被允许的是 ()。
- A. 使用 SSH 协议远程登录其它计算机以获取试题等文件
 - B. 写程序在评测环境中修改输入文件
 - C. 使用 U 盘拷贝题目及下发文件或自己的代码供赛后复盘
 - D. 通过枚举输入文件的可能情况获得答案并写入源代码

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 2 分, 选择题 3 分, 共计 40 分)

1.

```
01 #include <cstdio>
02 #include <cstring>
03
04 using namespace std;
05
06 char s[10000];
07 int cnt[26];
08
09 int main() {
10     scanf("%s", s);
11     for (int i = 0; i < strlen(s); ++i) {
12         if (cnt[s[i] - 'a'] <= 50) {
13             s[strlen(s)] = s[i];
14         }
15         ++cnt[s[i] - 'a'];
16     }
17     printf("%s\n", s);
18     return 0;
19 }
```

假设初始时输入的字符串长度不超过 500, 且不是空串。完成下面的判断题和单选题:

● 判断题

- 1) 将程序第 11 行中的“++i”改为“i++”, 程序运行结果不会改变 ()
- 2) 将程序第 11 行改为“for(int i=0, len=strlen(s); i<len; ++i)”, 程序的运行结果不会改变, 同时程序的运行效率将得到提升 ()
- 3) 对于任意一个出现了 a 到 z 中所有的字符、且各字符出现的次数不小于

50 的字符串 b , 总存在一个字符串 a , 使得将字符串 a 输入程序后的运行结果为字符串 b 。()

4) 程序的输出字符串长度一定不小于 1300 (注: $1300=50\times 26$)。()

● 单选题

5) 设输入字符串长度为 x ($1\leq x\leq 500$), 输出字符串长度为 y , 则关于 x 和 y 的大小关系正确的是 ()。

- A. 对于全部的输入字符串, 都有 $x=y$ 。
- B. 对于全部的输入字符串, 都有 $x<y$ 。
- C. 存在一部分输入字符串, 使得 $x=y$; 也存在一部分输入字符串, 使得 $x<y$; 但是不存在 $x>y$ 的情况。
- D. 存在一部分输入字符串, 使得 $x=y$; 也存在一部分输入字符串, 使得 $x>y$; 还存在一部分输入字符串, 使得 $x<y$ 。

6) (2分) 设字符串 w 为 $abcd\dots z$, 即从 a 到 z 在 w 中依次出现一次, 共 26 个字符。若输入为 w 重复出现两次的结果 (即 $abcdefg\dots zabcdefg\dots z$, 则输出结果为 ()。

- A. w 重复出现 50 次的结果。
- B. w 重复出现 51 次的结果。
- C. w 重复出现 52 次的结果。
- D. w 重复出现 53 次的结果。

2.

```
01 #include <cstdio>
02
03 const int N = 5010;
04 const int M = 20010;
05 const int inf = 1073741823;
06
07 int e, bg[N], nx[M], to[M], wt[M];
08 inline void link(int u, int v, int w) {
09     to[++e] = v;
10     nx[e] = bg[u];
11     wt[e] = w;
12     bg[u] = e;
13 }
14
15 int n, m, u, v, w;
16 int f[N], h[N << 1];
17
18 void update(int x, int y) {
19     x += n - 1;
20     for (h[x] = y; x; x >>= 1)
```

```

21     h[x >> 1] = f[h[x]] < f[h[x^1]] ? h[x] : h[x^1];
22 }
23
24 int main() {
25     scanf("%d%d", &n, &m);
26     for (int i = 0; i != m; ++i) {
27         scanf("%d%d%d", &u, &v, &w);
28         link(u, v, w);
29     }
30     int nn = n << 1;
31     for (int i = 1; i <= n; ++i) {
32         for (int j = 1; j != nn; ++j)
33             h[j] = 0;
34         for (int j = 0; j <= n; ++j)
35             f[j] = inf;
36         f[i] = 0;
37         update(i, i);
38         for (int j = i; true; j = h[1]) {
39             if (f[j] == inf) break;
40             for (int k = bg[j]; k; k = nx[k]) {
41                 if (f[j] + wt[k] < f[to[k]]) {
42                     f[to[k]] = f[j] + wt[k];
43                     update(to[k], to[k]);
44                 }
45             }
46             update(j, 0);
47         }
48         for (int j = 1; j <= n; ++j)
49             printf("%d%c", f[j], "\n "[j != n]);
50     }
51     return 0;
52 }

```

以下程序的输入是一张带边权的有向图，完成下面的判断题和单选题：

● 判断题

- 1) 将程序中所有的 “!=” 替换为 “<”，程序将仍然正常运行且输出的结果不会改变。 ()
- 2) 为了保证程序正常运行，输入的边数必须不大于 2×10^4 。 ()
- 3) 程序的输出是一个 $n \times n$ 的整数矩阵。 ()
- 4) 将程序第 34 行的 “j=0” 替换为 “j=1”，程序将仍然正常运行且输出的结果不会改变。 ()

● 单选题

- 1) (2分) 当输入的图中所有边的边权均为一个相同的正整数, 且有 $\sum w_i < 1073741823$ 时, “update” 函数被调用的次数为 ()。
- A. $\Theta(n^2)$
 - B. $\Theta(nm)$
 - C. $\Theta(n^2 + nm)$
 - D. $\Theta(n \min(n, m))$
- 2) 当输入的边权均为正整数时, 程序在最坏情况下的时间复杂度为 ()。
- A. $\Theta(n^3)$ 。
 - B. $\Theta(n^2 \log n + nm)$ 。
 - C. $\Theta(nm \log n)$ 。
 - D. $\Theta(n^2 m)$ 。

3.

```
01 #include <bits/stdc++.h>
02 using namespace std;
03
04 #define N 105
05 #define INF 1e9
06
07 int dis1[N][N], dis2[N][N];
08 int mp[N][N], n, m;
09
10 void fun1(int dis[N][N]) {
11     static bool vis[N];
12     for (int i = 1; i <= n; i++) {
13         for (int j = 1; j <= n; j++) {
14             dis[i][j] = mp[i][j];
15         }
16     }
17     for (int i = 1; i <= n; i++) {
18         for (int j = 1; j <= n; j++) vis[j] = 0;
19         for (int k = 1; k <= n; k++) {
20             int now = 0;
21             for (int j = 1; j <= n; j++) {
22                 if (!vis[j] && (!now || dis[i][now] > dis[i][j]))
23                     now = j;
24             }
25             vis[now] = 1;
26             for (int j = 1; j <= n; j++) {
27                 if (!vis[j] && dis[i][j] > dis[i][now] + mp[now][j]) {
28                     dis[i][j] = dis[i][now] + mp[now][j];
29                 }

```

```
30     }
31   }
32 }
33 }
34 void fun2(int dis[N][N]) {
35   for (int i = 1; i <= n; i++) {
36     for (int j = 1; j <= n; j++) {
37       dis[i][j] = mp[i][j];
38     }
39   }
40   for (int i = 1; i <= n; i++) {
41     for (int j = 1; j <= n; j++) {
42       for (int k = 1; k <= n; k++) {
43         dis[i][j] = min(dis[i][j], dis[i][k] + dis[k][j]);
44       }
45     }
46   }
47 }
48
49 int main() {
50   cin >> n >> m;
51   for (int i = 1; i <= n; i++) {
52     for (int j = 1; j <= n; j++) {
53       if (i == j) mp[i][j] = 0;
54       else mp[i][j] = INF;
55     }
56   }
57   for (int i = 1; i <= m; i++) {
58     int u, v, w;
59     cin >> u >> v >> w;
60     mp[u][v] = w;
61   }
62   fun1(dis1);
63   fun2(dis2);
64   int ans = 0;
65   for (int i = 1; i <= n; i++) {
66     for (int j = 1; j <= n; j++) {
67       if (dis1[i][j] != dis2[i][j])
68         ans++;
69     }
70   }
71   cout << ans << endl;
72   return 0;
73 }
```

以下程序的输入数据满足 $1 \leq n \leq 100, 1 \leq m \leq \frac{n(n-1)}{2}$ ，且只保证不存在重边，即不存在 $(u_i, v_i) = (u_j, v_j) (i \neq j)$ ，边权 $w_i \in [1, 10^6]$ 。如果 u 到 v 不可达，则认为距离为 INF 。完成下面的判断题和单选题：

● 判断题

- 1) 该代码的 $dis1[i][j]$ 不一定是 i 到 j 的最短路。()
- 2) 输出可能为 1。()
- 3) 将第 19 行的 $k \leq n$ 修改为 $k < n$ ，不影响答案。()
- 4) 对于稀疏图 (n, m 不同阶)， $fun1()$ 对于单个 i 求 $dis[i][j] (1 \leq j \leq n)$ ，最快可以做到 $\Theta((n+m)\log m)$ 。()

● 单选题

5) 对于以下的输入数据，输出结果为 ()。

```
5 8
3 2 2
2 4 2
1 4 3
3 1 2
4 3 3
5 2 3
1 5 1
1 2 2
```

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

6) 若输入数据 “n=5”，输出 ans 的最大可能值为 ()。

- A. 4 B. 5 C. 6 D. 7

三、完善程序 (单选题，每小题 3 分，共计 30 分)

1. (装备穿戴问题) 有 n 件装备，穿戴第 i 件装备需要玩家的力量值至少为 a_i ，穿戴该装备后会让玩家的力量值增加 b_i 。现在请问玩家的初始力量值最小是多少，才能以某种顺序穿戴上所有的装备？

输入：第一行是一个整数 $n (1 \leq n \leq 10^3)$ ；第二行有 n 个整数，第 i 个整数表示 $a_i (0 \leq a_i \leq 10^9)$ ；第三行有 n 个整数，第 i 个整数表示 $b_i (0 \leq b_i \leq 10^6)$ 。

提示：使用二分+贪心的方法解决这个问题，先对装备按顺序进行排序，然后二分答案，并贪心地进行选择。

试补全程序。


```

01 #include <cstdio>
02 #include <algorithm>
03
04 using namespace std;
05 const int maxn = 1005;
06
07 int n;
08 int a[maxn], b[maxn], c[maxn];
09
10 bool Comp(const int &x, const int &y) {
11     // 你可以简单地认为括号内的内容等价于 (int x, int y)
12     return ①;
13 }
14
15 bool check(int x) {
16     for (int i = 1; i <= n; ++i) {
17         int u = c[i];
18         if (②) {
19             x += b[u];
20         } else {
21             return false;
22         }
23     }
24     return true;
25 }
26
27 int main() {
28     scanf("%d", &n);
29     for (int i = 1; i <= n; ++i) scanf("%d", a + i);
30     for (int i = 1; i <= n; ++i) scanf("%d", b + i);
31     for (int i = 1; i <= n; ++i) c[i] = i;
32     sort(c + 1, c + 1 + n, Comp);
33     int ans = 1145141919;
34     for (int l=1, r=ans, mid=(l+r)/2; ③; mid=(l+r)/2)
35         if (check(mid)) {
36             ans = mid;
37             ④;
38         } else {
39             ⑤;
40         }
41     printf("%d\n", ans);
42     return 0;
43 }

```

- 1) ① 处应填 ()。
- A. $a[x] > a[y]$ B. $a[x] < a[y]$
 C. $a[x] \geq a[y]$ D. $a[x] \leq a[y]$
- 2) ② 处应填 ()。
- A. $x < a[i]$ B. $x < a[u]$
 C. $x \geq a[i]$ D. $x \geq a[u]$
- 3) ③ 处应填 ()。
- A. $l < r$ B. $l \leq r$ C. `check(l)` D. `check(r)`
- 4) ④ 处应填 ()。
- A. $r = mid - 1$ B. $r = mid + 1$
 C. $l = mid - 1$ D. $l = mid + 1$
- 5) ⑤ 处应填 ()。
- A. $r = mid - 1$ B. $r = mid + 1$
 C. $l = mid - 1$ D. $l = mid + 1$

2. (打音游) 小 A 最近喜欢上了一款音游, 并希望在结算时得到特定分数, 例如 1919810 分。这款音游的得分计算方式如下: 一共有 n 个音符, 将一千万 (10^7) 分平分给所有音符得到基础得分 $x = 10^7/n$ (保留非整数部分), 其中有 m 个音符根据是否击中可以获得 $x+1$ 分或者 0 分, 剩下的 $n-m$ 个音符根据击中精度可以获得 $x+1, x, x/2, 0$ 分中的一个, 最后将总分向下取整即可得到最终得分。
- 给定 n, m , 小 A 想知道他可以得到多少种不同的分数。
- 输入为两个非负整数, 分别表示 n, m ; 满足 $1 \leq n \leq 10^7, 0 \leq m \leq n$ 。输出为一个正整数表示答案。试补全程序。

```

01 #include<iostream>
02 using namespace std;
03 int main()
04 {
05     int n,m;
06     cin>>n>>m;
07     if(m==n) {
08         cout<<①<<endl;
09         return 0;
10     }
11     long long M=10000000;
12     int ans=②;
13     int lst=0;
14     for(int i=1;i<=n;++i) {
15         for(int j=1;j>=0;--j) {

```

```

16         int lower=max(0,③);
17         int upper=i-j;
18         int base=④;
19         ans+=upper-lower+1;
20         if(lower+base<=lst) ans-=lst-(lower+base)+1;
21         lst=⑤;
22     }
23 }
24 cout<<ans<<endl;
25 return 0;
26 }

```

- 1) ① 处应填 ()。
- A. -1 B. n-1 C. n D. n+1
- 2) ② 处应填 ()。
- A. -1 B. 0 C. 1 D. n
- 3) ③ 处应填 ()。
- A. $i - (n - m) - 1$ B. $i - (n - m) - j$
C. $i - (n - m)$ D. $i - (n - m) + 1$
- 4) ④ 处应填 ()。
- A. $(2*i+j) * M / (2*n)$
B. $(2*i-j) * M / (2*n)$
C. $i * M / n + j * M / (2*n)$
D. $i * M / n - j * M / (2*n)$
- 5) ⑤ 处应填 ()。
- A. base + upper B. base + upper + 1
C. base + lower D. base + lower + 1