2019CCF非专业级别软件能力认证第一轮

（CSP-J）入门级参考答案

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 |
| A | D | C | A | A | D | C | C | B | C |
| 11 | 12 | 13 | 14 | 15 |
| C | A | C | B | A |

二、阅读程序（除特殊说明外，判断题1.5分，单选题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)**（4分）** |
| × | √ | 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 |

**2019 CSP-J 第一轮 答案解析**

一：

1. .cn：中国的顶级域名是 .cn，概念性知识，但大家注册的时候其实对这个后缀非常熟悉了，rg.noi.cn
2. 01 0010 1000 0011：与运算是有 0 就为 0，都是 1 才是 1，所以比较快速的做法是看有没有 0。
3. 4：1 Byte = 8 bits
4. s=a-c：过去考过类似的，既然是做了 c 次 -1，自然等价于 -c。
5. 7：1(50),2(25),3(13),4(7),5(4),6(2),7(1)
6. 可随意访问任一元素：链表要访问到某个元素，必须一个个找过去。
7. 18：我们保证放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。8，17，26，35，44，116，125，134，224，233，1115....当然从这里我们也能发现递推方程。
8. 15：非常良心的考了原题，跟坐标是1，往右依次就是 3、7、15
9. 97：这个就没啥说的了。。
10. 29：如果辗转相除啥的可能会比较麻烦，最简单的方式就是验证一下四个选项是否除得尽。
11. 2400：阅读理解题，做法就按照说法来跑就好。
12. 4：抽屉原理，最坏情况就是最平均的情况，123412341234，还有一张必然会构成4。
13. 75：xyzyx 模式，其中 z 必须是(0,1,8)中的一个,y必须是(0,1,8,6,9)中的一个，前面是 6 后面就是 9，x 同 y，所以乘法原理后就是，3\*5\*5=75
14. ABDEGHJCFI：经典的内容，在后序遍历找根然后在中序遍历划分左右子树即可。
15. 图灵奖：果然有你

二 -（1）

1. x：输入的字符并没有什么限制。
2. √：我们下面索引用到了 i-1
3. x：在 i>sqrt(n) 时也是存在因子的
4. √：这个程序在做的就是特定位置小写转大写。
5. 6：只有 18 的因子的位置会被判断是否需要转换，共 6 个
6. 100000：验证每个选项，发现另外三个因子数明显没有36

二 -（2）

非常绕的一道题，也是让大家叫苦连天的题。到底在做什么呢？很快能发现 a 与 b 数组在做的是建立 x~y 的链接，设置的时候也是对称设置的。

如果 x 原来对应的 a[x] 比新的 y 小，并且 y 原来对应的 b[y] 比新的 x 小，那么把原来的对应关系清空，建立新的关系。

1. √：只要有关系来，不管会不会覆盖，至少不会关系全空。
2. x：看上去是对称的，但是因为有可能 x=1 对应着 y=2,那么做到 i==1 时 a[1]==2,b[1]==0;
3. x：如果我们建立了 2~3 与 3~2 两组关系明显就可以了。
4. x：15 行是在清楚原来 x 所对应的关系，和题目表述没啥联系。
5. 2n-2m：此时不会发生覆盖的情况，每组链接都被做了。
6. 2n-2：最后只会保留一组关系

二 -（3）

经验丰富的同学会看出来，这里就是在建立一颗树，每次找到当前区间最小值的位置作为根，然后划分左右做下一层。

最后的返回值非常重要，返回的是左边的值+右边的值+深度\*当前根权值，其实就是根节点权值为1，后续按照深度定权值

1. x：题目中找最小值有多个相等的话是随便选一个，不会有问题的
2. √：b数组是每个点的 value，a 数组是每个点的 key，最后计算是计算value\*depth，所以b数组全 0 结果就是全 0
3. 5000：每一层都要做一次，最坏情况下是一开始有序，每次都是第一个是根，所以要做 100 层，第一次 100 次，每层次数 -1，1+2+3+...+100 = 5050
4. 600：最好情况下就是每次都二分，层数应该是 log2(100)，也就是6左右。此时每层都差不多100个（后面随着有的点做了根会略少）。
5. 385：最坏情况就是1\*1+2\*2+3\*3....，手算一下，很快发现选择385.
6. 580：最好情况就是刚好每次都二分，那么 100 个节点的每层节点数量就是 （1,2,4,8,16,32,37）分别乘对应权值即可 1+4+12+32+80+192+259

三 -（1）

这题很明显是利用递归填好每个的位置，从最大的开始递归往下做。

1. t：x，y指当前的矩阵左上角，n是多少阶，那么t自然就是当前是 0 还是 1 了。
2. x,y：左上角矩阵
3. x+step,y+step：第二个是右上，第三个是左下，那么剩下的自然是右下了。
4. n,0：一开始的时候是 n 阶，0 为特征做下去。
5. 1<<n：最终大小自然是 2 的 n 次幂，这边复合的就是位运算的方式的结果。

三 -（2）

题目其实提示的非常充分了，空还是很好填的

1. ++cnt[b[i]]：题目说了，先按照b排序，那么自然是用b数组做关键字。
2. ord[--cnt[b[i]]] = i：理解这里最重要的是搞懂 18~19 行的内容， 18~19 行在处理我们上限范围内的前缀和，而其实本质上就是每个人的排名了。所以这边我们的cnt[b[i]]就是排名了，后续就很好理解了。
3. ++cnt[a[i]]：现在做a，同样的方式即可。
4. res[--cnt[a[ord[i]]]] = ord[i]：同上，此时我们采用 res 数组来处理最终结果。
5. a[res[i]],b[res[i]]：输出最终顺序的 a 与 b。