CSP-J 2021 初赛题详解

2021普及组真题

阅读程序部分难度比较高，其他的难度还可以，整体来说必之前难一点。

一、单项选择

1.D

2.B

3.A

4.C

​ N个数字找最大值，一般做法就是把第一个数字作为最大值直接存下来，然后和后面的每个数字比较，这种方法的比较次数是N-1,同时也是答案中的最小值了，故选 C。

5.D

模拟入栈出栈操作就可以。

6.D

​ 图有m条表，树的边数是n-1，即m - ( ) = n-1. 需要删除 m-n+1条边。

7.C

8.A

​ 高度为5的完全二叉树最后最后一层的节点数取值范围为1 - 16 故选 A

9.B

10.B

​ C62 \* C42 \* C22 / A33 先给每一组选人，因为不区分编号，消除因为组号带来的不同，故需要除以 A33

11.B

12.A

​ 分情况讨论：

​ 无重复数字：只能选123，然后A33

​ 有重复数字：先从1,2选出重复数字具体是谁，然后再选出唯一的数字，这样的三个数字有三种排列方法。

​ C21 \* C21 \* 3

​ 所以共有 A33 + C21 \* C21 \* 3 = 18种情况

13.C

​ 7 \* 5 \* 3 \* 2 \* 1

14.B

15.B

​ 这里过河有两种策略：

​ 策略1：让快的人送慢的人过河 然后快的人划船回来 可以节省回来的时间

​ 策略2：让两个慢的人一起过河 这样整体来说可能会节省一些时间

​ 过河过程：

​ 1 2 → 过河 ←1 回来

​ 4 8 → 过河 ←2 回来

​ 1 2 → 过河

​ 总的过河时间： 2 + 1 + 8 + 2 + 2 = 15

二、阅读程序

（1）

​ 题目重点是要知道 f(x) 和 g(x)的作用。

​ 首先看f(x)：

​ 假设 x 是 5,转换成补码进行运算

​ x： 0000 0000 0000 0101

​ x-1： 0000 0000 0000 0100

​ x&x-1： 0000 0000 0000 0100

​ x变成4:，继续运算

​ x： 0000 0000 0000 0100

​ x-1： 0000 0000 0000 0011

​ x&x-1： 0000 0000 0000 0000

​ 那么，不难看出，x&x-1的作用相当于去掉了二进制x中最低位的1，

​ f(x)这个算法叫做pop\_count()，作用是统计一个数字的二进制形式中有含有的 1 的数量。

​ 然后看g(x):

​ 假设 x 是 5,转换成补码进行运算

​ x： 0000 0000 0000 0101

​ -x： 1111 1111 1111 1011

​ x&-x: 0000 0000 0000 0101

​ 假设 x 是 4,转换成补码进行运算

​ x： 0000 0000 0000 0100

​ -x： 1111 1111 1111 1100

​ x&-x: 0000 0000 0000 0100

​ 那么，不难看出，x&-x的作用相当于取到了二进制x中最低位的1的权重。

​ g(x)其实就是树状数组中常用的 lowbit()操作。

错误

错误

错误

根据代码分析进行计算

2：2+1 = 3 正确

11：3+1 = 4 正确

9： 2+1 = 3 正确

16: 1+16 = 17正确

10: 2+2 = 4 错误

正确

直接计算：

511998 = 0111 1100 1111 1111 1110

16 + 2 = 18

20： 错误

需要有函数声明才行

B

2147483647这个数字可能大家会比较眼熟，他就是int的最大值 0后面跟31个1，所以第二个输出是31+1=32，故选B。

（2）

首先简单介绍下什么是 base64。

Base64就是用来将非ASCII字符的数据转换成ASCII字符的一种方法。特别适合在http，mime协议下快速传输数据。也可以用来加密，不过这种方法比较初级。

编码：

一个char在计算机中对应的是8个bit。

比如 'a' 在计算机中对应的 是97，对应的存储内容是0110 0001，占用8个bit。

这样的话 把三个char 对应的 24个bit 每6个一组变成4个元素，重新编码就达到了加密的目的。

解码：把四个元素转回三个字符。

1

2

3

4

5

错误

还可以出现其它字符。

正确

中间如果有换行符，可以使第二行相同 ，第三行不同。

正确

0xFF = FF(16) = 11111111(2) = char(-1)

即 int(char(-1))==-1 ? 不一定 和编译器有关 目前主流认为是正确的

25.B

B

代入计算即可

Y3Nx 每个字符对应6个比特位、

Y -> 24 -> 011 000

3 -> 55 -> 110 111

N -> 13 -> 001 101

x -> 49 -> 110 001

把数据拼接并重新编码：

01100011 01110011 01110001

99->c 115->s 113->q

C

方法同上。

（3）

这是一个欧拉筛法的代码。

a[] 标记数组 a[i]为false，表示i是质数。

b[] 存储所有质数

f[] 存储x的约数个数

g[] x的所有约数之和

28.正确

29.错误

30.错误

代数即可

31.A

32.C

约数个数为2，就是质数。

其实就是问有几个质数，100以内质数有25个

C

1000 = 23 \* 53

约数个数：

​ 幂次+1 的累乘： (3+1) \* (3+1) = 16

约数和：

​ 幂次对应等比数列和的累乘：（1+2+4+8）\* （1+5+25+125）= 2340

三、完善程序：

（1）约瑟夫问题

结合选项读代码，可知：

F[i] == 0表示没出圈。

单层循环：循环结束则任务完成，那么循环条件就和出队数有关

i是下标，c是剩余人数，还需要一个变量记录当前这个人报0还是1，这个应该就是p

D

筛掉n-1个人，任务结束，故循环的条件是 c<n-1

C

满足条件被筛掉 所以是判断 P==1 所以算 C

c

筛掉的人数+1

D

改变P的状态，下一个人应该报不同的数字。

B

无论是否被筛掉都需要进行的操作，就是让i从前往后周而复始的扫描。

（2）矩形计数

结合选项读代码，可知题目大致思路。

equals() 用来判断两个点是否相同，cmp()按照某种方式比较两点，sort（）冒泡排序，unique()用来去重，binary\_search()查找某个点点是否存在。

主函数读入所有点，排序去重。i,j分别是左上和右下两个点，然后在看是否存在对应的左下和右上的点，就可以知道能组成多少个矩形。

二分处的代码要一起分析，根据已有的二分的代码，可知空白4处条件是A[mid]小于p的情况，A[mid]大于等于p时 b=mid，相当于是找一个坐标第一次出现的位置，空白3处取mid时不用+1.

39.B

以x为第一关键字,y为第二关键字进行升序排列。

40.D

第一个点或者不重复的点，保留下来.

41.C

二分查找左边界，不用+1.

42.B

A[mid]小于p的情况，故A[mid]在前。

43.D

为了避免重复，必须前面的点坐标小于后面的点。