2021CSP-S1初赛答案评讲

一、单项选择题

1、A

ls：list列表，用于列出目录下文件和子目录。

cp：copy复制。

cd：change directory，切换目录。

all：linux中无此命令。

2、B

二进制运算，逢二进一。

3、A

概念题：系统栈又称调用栈，最经常被用于存放子程序的返回地址。

子程序被调用时，主程序必须将子程序运行完毕后应该返回到的地址暂存入调用栈，如果被调用的子程序还要调用其他的子程序，其自身的返回地址也必须存入调用栈，在其自身运行完毕后再行取回。

在递归程序中，每一层次递归都必须在调用栈上增加一条地址，因此如果程序出现无限递归（或仅仅是过多的递归层次），调用栈就会产生栈溢出，俗称“爆栈”，往往会引发不可预料的后果。

4、C

排序稳定的含义：相同元素在排序之后相对位置不会发生变化。

堆排序时会先将整个序列编成堆，之后每次将最大值移到最后，如存在两个最大值有可能会把前一个放到最后，所以不稳定。

5、C

思维题。2n个元素仿照堆的比较方式，两两比较n次（第1个和第2个比，第3个和第4个比…）分两组，胜者组找最大值n-1次，败者组找最小值n-1次，n+(n-1)+(n-1)=3n-2。

理解方法简单，但是作为选择题第5题需要学生在短时间内想出来还是有一定难度的。

6、C

开放寻址法（闭散列法）。一种解决散列冲突的方法（相对于分离链接法），适用于装填因子（散列表中元素个数和散列表长度）较小（小于0.5）的散列表。

本题手动模拟一下就行了，把0~7按照公式放到对应的位置，如果该位置已经有了，那就依次往后放。

模拟过程：0放0、1放1、2放4、3放9、4放5、5放3、6因为345都有了所以放6、7因为56都有了所以放7，答案选C。

7、C

简单图就是没有重叠和自环，即任意两点间只有一条边，这是一个不连通的有36条边的图，问有多少点？

如果有n个点，则最多包含数量是C n 2 C\_n^2C

n

2

​

，可以发现C 9 2 C\_9^2C

9

2

​

=36，即如果有9个点那就是一个完全图，必然连通，所以36条边答案是10个点（9个点的完全图+一个孤立点即可）

8、B

方法：列表找规律

高度 结点数

1 1

2 3

3 7

4 15

…… ……

10 2 10 2^{10}2

10

-1=1023<2021，不够

11 2 11 2^{11}2

11

-1=2047>2021，够了

所以是11层

9、D

本题暗含了一个信息：每个点的权值不一样。因为如果权值一样，那么不轮什么形态都一样。

前序遍历规则：根、左、右

中序遍历规则：左、根、右

不难看出如果相同，即根没有左子树，后面所有点都在右子树里，递归分析下去可以发现整个二叉树就是一路下去都只有右子树的一条链，所以是非叶子结点只有右子树的二叉树，选D。

10、A

交换一次的作用：减少一个逆序对，本题统计逆序对的数量即可：

DA DC DB CB FE FB EB

一共7个

11、A

递归函数：边界返回1，每次\*5%n

方法一：从t到1的过程中，每次乘以5，一共 t-1 次，所以本题即求5 t − 1 5^{t-1}5

t−1

%n，即求5 22 5^{22}5

22

%23，即2 5 11 25^{11}25

11

%23，根据同余定理等于2 11 2^{11}2

11

%23，即2048%23，计算结果为1

方法二：根据欧拉定理，522%23=1。因为5与23互质，而23的欧拉函数值为22，所以是1。本方法由gzx同学（数竞信竞双修大佬）提供，CSDN账号：gzx20210226，欢迎关注交流

12、C

递归计算斐波那契数列第n项，从n向下每个点都会再递归出两个点,1+2+4+…+2n-1，即一个二叉树，即O ( 2 n ) O(2^{n})O(2

n

)

13、C

经典奥数题

首先想到排列组合，每个苹果选不选一共是2 8 2^{8}2

8

，除去都不选的方案2 8 2^{8}2

8

-1，一共255种方案，再减去有两个苹果相邻的方案，此时发现似乎很难再往下想了；

换一种方法：动态规划，状态转移方程如下：

n个苹果的总方案数=第n个苹果不选的方案数（记作a） + 第n个苹果选的方案数（记作b）

而a应该与苹果数量为n-1的总方案数相同， b应该与苹果数量为n-1时第n-1个苹果不选的方案数相同

手动模拟过程如下：

苹果数量n 1 2 3 4 5 6 7 8

第n个苹果选的方案数a 1 1 2 3 5 8 13 21

第n个苹果不选的方案数b 1 2 3 5 8 13 21 34

n个苹果的总方案数=a+b 2 3 5 8 13 21 34 55

最后再除去什么都不选的方案数55-1=54

14、C

组合数学问题（热门考点）

一共9 3 9^39

3

=729种方案，排除非法方案，有两种情况：

1、三个数均不一样，987=504

2、两边相等中不能构成三角形（两边之和小于等于第三边）的方案，包括从112、113、114……至338、339、448、449，共8+6+4+2=20种，每种组合对应三种排列（如112有112、121、211三种情况），20\*3=60

最后计算729-504-60=165，选C

15、B

ACEHJ=19，手动模拟下就行，也可以用dp、dijkstra来做，但没必要，这张图随便看看很快就有答案了

二、阅读程序

阅读程序题方法：首先看问题，问题中可能会有题目的提示，可以帮助我们更快读懂题目

第一个程序：√ × × √ D C

计算几何题

acos是cos的反函数，cos(π/3)=0.5，所以acos(0.5)就是π/3

t=|x1-x2|2+|y1-y2|2+|z1-z2|2

可以估计一下，abc应该是点，d可能是半径

代码中有“cout<<0”，会输出0一定不可能是并集，故21题只能选C：求球的体积交

进一步分析发现a1、b1、c1应该是第1个球的中心点坐标，d1是半径，a2、b2、c2应该是第2个球的中心点坐标，d2是半径

此时基本可以理解代码所表达的算法思路，不难分析出第一个分支表示球的包含关系，第二个分支表示相离，第三个分支表示相交，并不需要去管第三个分支里的代码证明与含义，照做即可

16、17、18看代码：

16题描述正确。

17题错误，交换之后会变成整除，参考3/2/1.0与3/1.0/2的区别 。

18题，x为double类型，里面存放的值可能为XX.5，而sq(const int x)会先将实参取整，与 x \* x 结果不同，故错误。

19题，判断发现两球相交，走第三个分支，带入公式去算即可，计算略繁，算出来答案1.30899，精确4位后即1.3090。正确。

20题，首先发现球心重合，所以体积就是4 3 \frac43

3

4

​

π r3,r是1，所以就是3.14159\*4/3=12.56636/3=4.18878≈4.1888

关键：通过最后一题理解代码含义后本题就很好做了

第二个程序：√ × × B C B

本题需花费较长时间观察，首先看一下问题，第22题明显是出题人好心提醒，让我们先阅读一下两个函数，题意就是看一下 solve1(1,n).j 和 solve2(1,n) 是否等价，后面几个问题倒暂时看不出什么有用信息。

之后查看代码整体框架，开头定义了一个结构体，结构体里面重载了运算符+；之后是两个递归函数，主函数输入n个整数，之后两个函数各调用了一遍。

先看第一个函数，运算符重载，代码比较乱，看不懂正常，接下来就来看第二个函数吧：

发现两个递归函数的边界一样

有一个区间[h,m]，中点是j（>>1的含义就是/2）

第一个for循环的含义：，wht用来算后缀和，wh求后缀和的最大值；

同理第二个for循环：wmt用来计算前缀和，wm是前缀和的最大值。

最后发现递归的返回是前半段、后半段、wh+wm中的最大值，那其实就是求一个区间中的最大连续子段和。

第二个函数就是暴力求解一个区间中的最大连续子段和的方法。

再看就发现第一个函数是用线段树实现两个区间合并，线段树需要维护（也可以用splay来维护）4个信息：最大连续子段和、最大前缀和、最大后缀和、当前区间的总和，再将结构体中四个变量hjmw和上述四个对应起来理解程序代码即可。

最大连续子段和 最大前缀和 最大后缀和 当前区间的总和

j h m w

第23题，第28、38行的代码就是返回区间为空，那么区间为空会执行两次以上吗？首先看输入范围，输入的值不超过1000，那么可以有0和负数，如果有负数代码只会执行一次。那如果是正数是否会出现区间长度为-1？显然不可能，可以反推。错误。

第24题，单单11就比7大，错误

第25题，solve1为递归函数，每个区间分别递归，每次递归区间长度减半，第一层2个点，第二层4个……，最后一层n个点，一共创建了2n个点，时间复杂度为O(n)。选B

第26题，与上题一样，共递归 log n 层，根据solve2代码，每一层都要线性循环一遍复杂度O(n)，n\*logn，选C。

第27题，注意第一个10是n，不是序列，选C的同学看清题目。

本题是求最大连续子段，需要先修线段树

第三个程序（手算base64）：× √ × B D D

base64是一种加密算法如下：

加密：

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。

解密：

把四个元素转回三个字符。

1

2

3

4

5

6

7

encode、decode，根据函数名称及代码中的ASCII码值得知为加密解密，table用来求每一个字母在base中的下标。

代码难度略大，可代入样例跑一下。

第28题，以为答案错了，大佬给的解释是因为第二个cout最后有endl换行到第3行，第3行为空，所以应该是3行而不是两行。所以错误。

听完以后的反应就是？？？？？？？？？？？？？？？？？？？？？？？？？？？？？？？？？？？？ 毒瘤题鉴定完毕

第29题，由算法可知加密解密唯一对应，所以正确。

第30题，“SGVsbG93b3JsZA==“这么长一串我打一遍都嫌累，还要手算？？？太麻烦了，建议瞎蒙一个，鉴于本试卷的难度以及字符中的第4个和第5个故意小写嘲讽的字符，我给×评。换个思路，让我们来猜猜出题人的想法：“如果是对的，那么考生需要把这么长一串算完才能知道是对的，他们算完肯定会骂我，还是做个人吧，让他们算到一半发现是错的就不用往下算了！”

好吧，真要算的话其实有一个简单的方法，先看一下最后的”=="，看看到底是不是两个"="，结果数完等号就发现错了所以×。本方法由cly同学（我校信竞队队长）提供，CSDN账号：XL4453，欢迎关注交流

第31题，每个元素只用了一遍，O(n)

第32题，oxff是-1，选D

第33题，0表示加密encode，CSP2021csp一共10个字母编码，10%3=1，根据第34行代码会补两个"=="，所以BD中选，发现BD大部分相同，只有第三组第二个字母（G、W）不一样，G、W在base中下标分别为（000110）和（001100），看是对应位置01还是11就行了，把‘c’（ASCII码为99）转成二进制110 0011，前两位是11对应W，选D。

三、完善程序

第一个程序：D A D C

使用了类似Dijkstra的思想，设F[i]表示用4表示i最少需要多少个，显然F[4]=1。如果已经确定了一个数字x的F[x]，就可以更新任意其他数字的F，比如F[i+x]=F[i]+F[x]，因此每次取出F最小的数字x，尝试用x更新所有未确定的数字。之后F[x]的值肯定无法继续更新了，也就被确定了，因此不应该再访问x，把对应的vis[x]赋值为1，这个动作可以一直进行，知道F[n]的值被确定了。

34、算法的起点是4，所有数字都要用4拼出来，而F[4]=1。

35、如果F[n]的值被确定了，那么不需要再继续计算了。

36、根据Dijkstra思想，已经取出的不再取出，应该取出未取出的点里面F最小的。

37、只使用确定了最小值的点来更新其他点，如果去掉这个if语句也不影响程序的正确性。

第二个程序：A D A D D C

程序算法是使用笛卡尔树解决RMQ问题，首先建立笛卡尔树，然后使用DFS求欧拉序，再求分块后块与块之间的最小值，最后解决块内最小值，完成预处理工作后依次回答输入的询问。

笛卡尔树同时满足两个性质，从key来看，满足二叉搜索树的特性，从value来看，满足堆的性质，笛卡尔树的建树过程可以采用增量法：

我们考虑将元素按照键值k排序，然后一个一个插入到当前的笛卡尔树中。那么每次我们插入的元素i必然在这个数的右链的末端（从根节点一直往右子树走到底）。这里采用单调递增栈来构建笛卡尔树，栈中维护当前笛卡尔树的右链上的节点。每次插入当前的i，在单调栈中不停地弹出栈顶，直到栈顶fa满足val[fa]<val[i]，把最后一次弹出的设为j。将i作为fa的右儿子，j作为i的左儿子即可。

38、对应建立笛卡尔树时，将j作为i的左儿子

39、i作为fa的右儿子

40、在树上求lca，因此返回更浅的结点。

41、笛卡尔树的性质保证，深度越浅的点val越大

42、如果mask的第i-1是1，代表a[i]=a[i-1]-1

43、从89行可以看出，在mask中，较低位在二进制的左边，因此右移的数量与l有关

四、初赛试卷整体难度评析

今年的初赛试卷整体看下来很难，很明显是故意人为增加了难度：手算6位浮点小数（进位、计算几何、acos(0.5)）、手算base64、用dijkstra跑可以直接输出的答案、四毛子树……加上现在为了降低批改试卷的难度全是选择判断题导致运气太差便可能进不了复赛了，算是近几年最难的一次了，但是拥有复赛水平的学生做个40左右难度也不算大，再蒙个几题就能进了（选C的26分+判断题15分得个一半7.5分=33.5分，所以这试卷闭眼2分钟时间就可以做出33.5，考20分的是怎么做到的，不会的就选C这句话没听过？），江苏除了分数线极高的个别市，大部分该进复赛的选手都进复赛了，倒也不必怨天尤人哀鸿遍野。

建议同学们初赛还是不要裸考，起码把往年题目看一看，看有没有哪块知识点盲区，查漏补缺，当然也不建议花费超过20h准备初赛，影响了备战复赛反而得不偿失，毕竟准备复赛相对地也会提高初赛成绩，例如今年的线段树、base64、dijkstra、笛卡尔树等，如果考生在之前没有写过有关程序那大概率是做不出来的。

————————————————