



# CSP-J 2022 第一轮 真题讲解

LeavingZ 洛谷网校

# 单项选择题

---

面向对象/面向过程：

对象指的是具体的某个事物，在编程语言中包括数据和动作（可以理解为成员变量和成员函数），面向对象的显著特点包括封装和继承。

封装：将数据和函数代码捆绑到某个对象上，访问封装的内容只能通过已定义的一些方式，利用修饰符，保障了对象内容的独立性和安全性。

继承：我们可以新建一个子类（又叫派生类），让其继承已有的某个类（被继承的类称作父类/超类/基类）的特性，我们可以在继承的特性之上扩充或者改写一些内容。

与类/结构体紧密相关

# 单项选择题

---

面向对象/面向过程：

过程可以理解为一个事件，我们为了实现某个功能，将这个功能划分为几个依次发生的事件（过程），然后用函数依次实现这些过程，然后按照顺序调用这些函数就可以实现这个功能。

与函数紧密相关

# 单项选择题

---

某个数字出现在出栈序列中，那么在这个数字之前进栈的数字，要么已经出栈，要么按照顺序留在栈中，我们可以模拟四个选项。就可以知道哪个序列非法。

栈-先进后出

# 单项选择题

`int x;` 是声明一个标识符为 `x` 的整型变量。

`&x` 是取变量 `x` 在内存中的地址。

`int *p;` 是声明一个标识符为 `p` 的指向整型变量的指针。

指针存储的是一个地址，因此 `int *p=&x;` 表示让 `p` 指向 `x` 的地址。

那么 `p = q;` 就是把 `q` 存储的地址赋值给 `p`，此时 `p` 中就会存储变量 `y` 的地址，所以 `p` 指向 `y` 的地址。

此外，`*p` 表示访问指针 `p` 指向的地址的值（类型+地址，可以在内存里取出对应大小的数据）。

# 单项选择题

## 数组 vs 链表

数组：连续的确长度的一系列同类型元素。

链表：依靠节点间的链接形成的一条链。

数组的每个单位：存储这个单位的信息。

链表的每个单位：存储这个单位的信息和下一个节点的链接。

因此，数组可以利用下标随机访问（内存上连续，可以由首地址和下标直接推断出存储位置），但是链表不可以（依赖每个节点的链接进行跳转，链接没有规律）；同时链表可以增加/删除元素（添加/删除节点和链接即可），但是数组不可以（长度确定无法变化）。

## 单项选择题

---

队列先进先出，因此队列的出队列顺序就是进队列顺序，元素不进队列就会留在栈中，同时只要进队列就一定经历了进栈和出栈，所以其实是告诉了我们进栈顺序和出栈顺序。

我们要求栈  $S$  的最小容量，那么对于每个元素，只要能不进栈就暂时不进，只要可以出栈就立即出栈，这样可以最小化栈中的元素数量，栈的容量其实就是求栈中元素数量的最大值。

# 单项选择题

## 中缀表达式转前/后缀表达式

先来看最基础的一种， $a+b$  转前缀，那么就是  $+ab$ 。

那么复杂的呢，例如题目的  $a+(b-c)*d$ ，我们可以把后面的  $(b-c)*d$  看作一个整体，那么就是  $+a (b-c)*d$ ，显然  $(b-c)*d$  还需要进一步转化，那么就把  $(b-c)$  看作一个整体，写成  $*(b-c)d$ ，然后把  $b-c$  转化为  $-bc$ ，合起来就是答案  $+a*-bcd$ 。

后缀表达式也类似，把符号放在数字后面就是了。

$a-b-c$  的前缀表达式是  $--abc$ ，对于讲先后关系的运算符同等级的我们用最后一个 先变成  $-(a-b)c$  再变成  $--ab c$



# 单项选择题

---

## 哈夫曼编码

每次选择出现频率最低的两个字符，将其合并，直到只剩下一种字符，形成的二叉树结构即为哈夫曼树，二叉树的每个叶子代表了一种字符，这个叶子的深度即为其编码长度。

## 二叉树

完全二叉树从 1 开始的编号方法：节点  $i$  的左子结点编号为  $2*i$ ，右子结点的编号为  $2*i+1$ ，所以所有的右子结点编号都是奇数，其兄弟节点就是自己的编号  $-1$ 。

# 单项选择题

## 图论

有向图邻接矩阵的非零元素最少个数=最少的边数。

也就是问你最少要多少条边能让这个有向图联通，最好的方法是让  $N$  个节点首尾相连构成一个环，即为一个强连通分量。

深度优先搜索-递归-栈-后进先出

广度优先搜索-队列-先进先出

用两个栈也可以实现队列。把入队的操作都放进一个栈，要出队的时候就出第二个栈的栈顶，如果第二个栈是空的就把第一个栈的元素依次出栈加入到第二个栈中，再取栈顶出栈即可。

# 单项选择题

## 双向循环链表

赋值的时候注意不要使用被覆盖过的值。分析我们要修改的变量以及要让其变为什么，判断出先后顺序，后面要用到的变量不能先修改。

排序的稳定性：相同的元素先后顺序会不会改变，会改变那么不稳定。

冒泡排序：相同的元素要不同时被左移要不同时被右移，稳定。

选择排序：相同的元素中靠前的元素可能会被交换到后面，不稳定。

插入排序：相同的元素插入时不会破坏先后顺序，稳定。

归并排序：合并左半部分和右半部分的时候值相同的时候优先使用左半部分的元素可以达到不破坏先后顺序的目的，稳定。

# 单项选择题

---

## 进制转换

K 进制转 10 进制，分小数点前后进行转换。

$$(32.1)_8 = 8^1 \times 3 + 8^0 \times 2 + 8^{-1} \times 1 = 26.125$$

## 递归-函数调用自己

# 阅读程序

---

带符号整型溢出是未定义行为。

char/bool 1 Byte

short 2 Bytes

int 4 Bytes

long long/float 8 Bytes

double 16 Bytes

位运算，模拟即可，注意把 16 进制转为 2 进制运算。

# 阅读程序

## 递归变递推

研究发现，这其实是同一个计算过程的递推和递归实现（理由：边界情况相同，范围相同，递推式子完全一致）。

循环次数（或者 min 函数的执行次数）也可以由递归变递推。

$p[n][m]$  在  $n = 0$ ，或者  $m = 1$  的时候为 0，其他时候按照递归的方程，先让  $p[n][m] = n$ ，然后枚举  $i$ ， $p[n][m] += p[n-i][m] + p[i-1][m-1]$ 。通过这种方式变为递推，然后就可以在草稿纸上算出  $p[7][3]$  了。

递推式子容易发现，第一个下标之和为  $i-1$ ，第二个下标即为当前下标和当前下标减一（当前列和前一列），找规律。

## 阅读程序

---

solve1 是整数二分开根号。

solve2 是迭代求近似根号。

$(x - \sqrt{n})^2 \geq 0$  可以得到  $\frac{x^2+n}{2x} \geq \sqrt{n}$ , 同时也可以由  $x \geq \sqrt{n}$  得到  $\frac{x^2+n}{2x} \leq x$ , 因此  $x$  会逐步靠近  $\sqrt{n}$ 。

# 完善程序

算法流程：

枚举  $1 \sim \sqrt{n}$ （不包括  $\sqrt{n}$ ）的所有整数因子，如果能够整除那么加入到 fac 中，特别判断  $n$  是不是完全平方数（防止平方因子输出两次），然后根据其他因子成对出现的特性输出剩下的因数。



# 完善程序

---

## 算法流程

从点 `cur` 出发广度优先搜索，每次检查当前点的周围点是不是合法的点，合法性的检查包括是否越界（到图像之外），是否和起始像素颜色相同，与是否已经染成给定颜色。合法的点进行染色，并且加入队列继续扩散寻找。