

# 算法

**本题库可用来参考复试出题范围，答案可能存在一些问题，感谢群友们随时指出~**

## 1、分支定界和回溯法区别。

答：(1) 求解目标不同：回溯法的求解目标是找出解空间树中满足约束条件的所有解，而分支限界法的求解目标则是找出满足约束条件的一个解，或是在满足约束条件的解中找出在某种意义下的最优解。

(2) 搜索方式的不同：回溯法以深度优先的方式搜索解空间树，而分支限界法则以广度优先或以最小耗费优先的方式搜索解空间树。

在分支限界法中，每一个活结点只有一次机会成为扩展结点。活结点一旦成为扩展结点，就一次性产生其所有儿子结点。在这些儿子结点中，导致不可行解或导致非最优解的儿子结点被舍弃，其余儿子结点被加入活结点表中。此后，从活结点表中取下一结点成为当前扩展结点，并重复上述结点扩展过程。这个过程一直持续到找到所需的解或活结点表为空时为止。

## 2、\*\*什么是NP问题？（\*超纲题目\*）

答：P类问题：所有可以在多项式时间内求解的判定问题构成P类问题。判定问题：判断是否有一种能够解决某一类问题的能行算法的研究课题。

NP类问题：所有的非确定性多项式时间可解的判定问题构成NP类问题。非确定性算法：非确定性算法将问题分解成猜测和验证两个阶段。算法的猜测阶段是非确定性的，算法的验证阶段是确定性的，它验证猜测阶段给出解的正确性。设算法A是解一个判定问题Q的非确定性算法，如果A的验证阶段能在多项式时间内完成，则称A是一个多项式时间非确定性算法。有些计算问题是确定性的，例如加减乘除，只要按照公式推导，按部就班一步步来，就可以得到结果。但是，有些问题是无法按部就班直接地计算出来。比如，找大质数的问题。有没有一个公式能推出下一个质数是多少呢？这种问题的答案，是无法直接计算得到的，只能通过间接的“猜算”来得到结果。这也就是非确定性问题。而这些问题的通常有个算法，它不能直接告诉你答案是什么，但可以告诉你，某个可能的结果是正确的答案还是错误的。这个可以告诉你“猜算”的答案正确与否的算法，假如可以在多项式（polynomial）时间内算出来，就叫做多项式非确定性问题。

## 3、n个活动，第i个活动的开始结束时间是 $[s_i, f_i]$ ，每个活动排他性的占用场地，如何安排才可以尽可能安排更多的活动，求解算法

贪心算法：对活动以其完成时间的非减序排列

设待安排的11个活动的开始时间和结束时间按结束时间的升序排列如下：

i: 1 2 3 4 5 6 7 8 9 10 11

s[i]: 1 3 0 5 3 5 6 8 8 2 12

f[i]: 4 5 6 7 8 9 10 11 12 13 14

分析：每个活动*i*都有一个要求使用该资源的起始时间 $s_i$ 和一个结束时间  $f_i$ ，且 $s_i < f_i$ 。如果选择了活动*i*，则它在半开区间 $[s_i, f_i)$ 内占用资源。若区间 $[s_i, f_i)$ 与区间 $[s_j, f_j)$ 不相交，则称活动*i*与活动*j*是相容的。也就是说，当 $s_i \geq f_j$ 或 $s_j \geq f_i$ 时，活动*i*与活动*j*相容。

例：给出待安排的11个活动的开始时间和结束时间，要求安排尽量多项活动使用会场。

首先，任意输入这11个活动。

然后**对活动以其完成时间的非减序排列**。（意义：使剩余的可安排时间段极大化，以便安排尽可能多的相容活动。）

将第一次活动结束时间 $f[1]$ 与后面活动开始时间 $s[2]$ 相比较，若 $s[2] < f[1]$ 则继续比较，直到 $s[4] > f[1]$ ，选中此活动。再用活动4的结束时间 $f[4]$ 与其后活动的开始时间比较……同理类推，直到比较完成为止，最后选出符合条件的活动1，活动4，活动8和活动11，它们将依次被安排使用该场地。

#### 4、给你一个指针数组p和一维数组a构造一个二维数组，并且按行和按列分别输出

让指针数组每个元素按照固定间隔存储一维数组元素的地址即可

#### 5、给了一个单链表，说要把单链表循环右移若干个位置

比如123456变成561234

首先计算出位移是否出现循环，即用位移数mod链表长度，例如位移数mod链表长度为n，则将链表后n个元素插入到链表的头部

#### 6、C++的三种继承方式

public、protected、private三种继承方式，相当于把父类的public访问权限在子类中变成了对应的权限。

#### 7、c语言实现strcpy函数功能

功能描述：strcpy能实现将某一个数组字符串拷贝到另外的一个数组中实现字符串的复制。

首先清空数组1，然后写一个循环，将数组2的元素一个一个复制到数组1中。最后加上'\0'。

#### 8、题目：老师学生吃饭，老师和学生吃的不同，就需要定义两个类。

问题一：怎样定义这两个基类？（貌似是的，就是基类问题）

问题二：用eat函数调用实现这两个基类？

## 9、说明一下大O记号

大O符号（Big O notation）是用于描述函数渐进行为的数学符号。更确切地说，它是用另一个（通常更简单的）函数来描述一个函数数量级的渐近上界。在数学中，它一般用来刻画被截断的无穷级数尤其是渐近级数的剩余项；在计算机科学中，它在分析算法复杂性的方面非常有用。

## 10、贪心算法和动态规划的区别？

答：贪心算法的基本思想：贪心算法并不从整体最优上加以考虑，它所做的选择只是在某种意义上的局部最优解。基本要素：最优子结构性质和贪心选择性质。

动态规划的基本思想：将待求解的问题分解成若干个子问题，先求解子问题，然后从这些子问题的解得到原问题的解。基本要素：最优子结构性质和重叠子问题性质

相同点：两者都具有最优子结构性质。

不同点：动态规划算法中，每步所做的选择往往依赖于相关子问题的解，因而只有在解出相关子问题时才能做出选择。而贪心算法，仅在当前状态下做出最好选择，即局部最优选择，然后再去解做出这个选择后产生的相应的子问题。

## 11、n个数（n为偶数），比较次数不超过 $(3/2)n-2$ 。找出最大和最小的数。

答：第一趟遍历，每两个数一组进行比较，较大的数放在奇数（或偶数）位上。比如第一个数和第二个数比较，第三个数与第四个数比较...这样经过一趟遍历（比较次数 $n/2$ 次），最大的数一定是在奇数位置上，最小数一定是在偶数位置上。于是进行第二趟遍历，类似于在偶数位置和奇数位置上分别进行一趟冒泡排序（比较次数为 $(n/2 - 1) + (n/2 - 1)$ ），这样，在倒数第二个位置上必然是最大数，在最后一个位置上必然是最小数。总的比较次数为 $(3/2)n - 2$ 次。

## 12、prim算法生成最小生成树过程，为什么一定是最小生成树？

答：设图为 $G(V, E)$ 有 $n$ 个结点，Prim算法的做法是，初始化一个空点集 $S$ 和边权之和 $ans = 0$ ，进行 $n$ 次循环，每一次循环都从未被加入 $S$ 的结点中选取一个到 $S$ 距离最小的结点 $u$ 并加入 $S$ ，并将 $u$ 到 $S$ 的距离累加到 $ans$ 上。若在某一次循环中未找到这样的结点 $u$ ，则说明该图为非连通图，则不存在生成树。否则，循环结束后， $ans$ 记录的就是最小生成树的边权之和。

Prim算法和Kruscal算法之所以是正确的，是因为它们都运用了最小生成树的一个性质，即假设 $G(V, E)$ 是一个连通图， $U$ 是顶点集 $V$ 的一个非空子集，若 $(u, v)$ 是一条具有最小权重的边，其中 $u \in U, v \in V - U$ ，则必存在一颗包含边 $(u, v)$ 的最小生成树。这个性质可以通过反证法得到证明。

## 13、小明有十个小时空闲时间，现在有一批兼职可以做，每个兼职有开始时间 $s$ 结束时间 $t$ ，问找出一个算法计算出小明的兼职安排使小明的兼职数量最多。

问题可以转化为区间不相交问题：给出N个开区间(x, y)，从中选择尽可能多的开区间，使得这些开区间两两没有交集。贪心解法：总是先选择左端点最大的区间（或者是右端点最小的区间）

详细解答算法思路：

[https://blog.csdn.net/weixin\\_43553694/article/details/104221771](https://blog.csdn.net/weixin_43553694/article/details/104221771)

#### 14、将多个名字用什么结构存储，可以通过scanf读取名字的首字母

枚举类型

#### 15、给定一个整数数组和一个整数 k，判断数组中是否存在两个不同的索引 i 和 j，使得 $\text{nums}[i] = \text{nums}[j]$ ，并且 i 和 j 的差的绝对值至多为 k。

思路：将每个元素与它之前的 k 个元素中比较查看它们是否相等。

```
public boolean containsNearbyDuplicate(int[] nums, int k) {  
    for (int i = 0; i < nums.length; ++i) {  
        for (int j = Math.max(i - k, 0); j < i; ++j) {  
            if (nums[i] == nums[j]) return true;  
        }  
    }  
    return false;  
}
```

时间复杂度分析

时间复杂度： $O(n \cdot \min(k, n))$

每次搜索都要花费  $O(\min(k, n))$  的时间，哪怕 k 比 n 大，一次搜索中也只需比较 n 次。

空间复杂度： $O(1)$

#### 16、两个数组，找交集，并说明时间复杂度

初始化flag数组为0。对A数组的第i个值，将flag[A[i-1]]+1。判断flag[B[k]]是否为空，不为空则把此值加入并集。时间复杂度为  $O(n)$ 。

#### 17、类型检查和类型转换

1、编译器只对确定类型进行检查；

2、类型转换是否能够成功有没有明确的判断；编译时或运行时；

3、任意类型不参与编译时类型检查；任意类型不能直接参与运算，必须转化为确定的类型；任意类型转化为确定类型是否需要显式转化。

强类型：类型检查（编译时）和转化（运行时）都有明确的状态信息；

中类型：类型检查（编译时）强制，转换无明确状态信息；

弱类型：全部无；

动态类型：

任意类型变量中包含类型信息，可以在运行时获取；

继承类型（动态）：变量中包含类型信息；

**18、请说一下如何实现电文加密，历年真题里的原题，加密规则是A转换为Z，B转换为Y，C转换为X，a转换为z，b转换为y，c转换为x**

通过判断其ASCII码，可以确定其为大写或者小写字母。例如其为小写字母，则

```
int result = 'a' + 'z' - x;
```

### 19、怎么理解抽象数据类型

抽象数据类型是一个数学模型以及定义在其上的一组操作组成，因此，抽象数据类型一般通过数据对象、数据关系以及基本操作来定义，即抽象数据类型三要素是(D, R, P)。

20、数组与结构体的区别和联系

- 抽象数据类型的特征主要体现在以下几个方面：[1]
- **数据抽象**。用ADT描述程序处理的实体时，强调的是其本质的特征、其所能完成的功能以及它和外部用户的接口（即外界使用它的方法）。
  - **数据封装**。将实体的外部特性和其内部实现细节分离，并且对外部用户隐藏其内部实现细节，它包含两层含义：
    1. 将数据和其行为结合在一起，形成一个不可分割的独立单位；
    2. 信息隐藏，即尽可能隐藏数据内部细节，只留有限的对外接口形成一个边界，与外部发生联系。封装的原则使得软件错误能够局部化，大大降低排错的难度，便于软件的维护。
  - **继承性**。**数据封装**使得一个类型可以拥有一般类型的数据和行为，即对一般类型的继承。若特殊类型从多个一般类型中继承相关的数据和行为，则为**多继承**。
  - **多态性**。**多态性**是指在一般类型中定义的数据或行为被特殊类型继承后，具有不同的数据类型或呈现出不同的行为。例如，“苹果”是“水果”的子类，它可以有“水果”的一般“吃”法，但其本身还可以有别的多种“吃法”。

**20、两个进程并发执行 $p=30$ ,  $T1\ p=p+10$ ,  $T2\ p=p+p*0.3$ ，得出结果和过程。**

并发执行有两种可能

1. 先 $T1$ 后 $T2$ :  $p=30+10=40$ ;  $p=40+40*0.3=52$

2. 先 $T2$ 后 $T1$ :  $p=30+30*0.3=39$   $p=39+10=49$