**算法设计与分析往年试题答案**

1. 判断题
2. 一个正确的算法，对于每个合法输入，都会在有限的时间内输出一个满足要求的结果。

对。

1. NP完全问题比其他所有NP问题都要难。

NP问题分为NP-hard问题和NP完全问题，NP完全问题是最难的，但是NP-hard问题又包含NP完全问题。

错：NP完全问题至少同其他所有NP问题一样难。

1. 回溯法用深度优先法或广度优先法搜索状态空间树。

错：回溯法用深度优先法搜索状态空间树。

1. 在动态规划中，各个阶段所确定的策略就构成一个策略序列，通常称为一个决策。

错：在动态规划中，各个阶段所确定的决策构成一个决策序列，通常称为一个策略。

1. 类和类问题的关系用来表示是错误的。

错：P中所有问题均属于NP。

P：存在求解判定问题P1的多项式时间算法。

NP：对P1的每一个肯定实例均存在一个多项式时间内的验证。（对一个实例为肯定实例的证明称为该实例的证书。）

NPcomplete:P1是一个NP问题，且NP中所有问题都可以多项式转化为P1.

NPhard：NP中所有问题都可以多项式转化为P1。

NPhard问题范围大于NPcomplete问题范围。

1. **若近似算法A求解某极小化问题一实例的解为，且已知该问题的最优解为，则该近似算法的性能比为3。**

**错：**性能比是所有实例可能的精确率的上界。3只是这一实例的精确率，不是所有实例的。

1. 通常来说，算法的最坏情况的时间复杂行比平均情况的时间复杂性容易计算。

对。

1. 若P2多项式时间转化为(polynomial transforms to) P1，则P2至少与P1一样难。

错：应该是p1至少和p2一样难，有可能p1更难。

总是可以在可比时间内用P1的算法解决P2，但不能说P1和P2一样难。事实上，有时候可能P2简单而P1更难。

1. 快速排序算法的平均时间复杂度是，使用随机化快速排序算法可以将平均时间复杂度降得更低。

错。

1. **基于比较的寻找数组中最大元素的问题下界是。**

**对**：下界理论：问题的下界不唯一，越高越好。算法最优：下界=上界

几个问题的下界：

1. 排序问题：
2. 找最大：或
3. 找最大最小： 下界是用分治策略得到的结果：将数组均分成两部分，在每部分中找出最大值和最小值，在比较这两部分的最大值和最小值。
4. 找第二大元素： 找第一大元素需要比较n-1次，第一是通过两两配对比较中较大的元素中最大的元素，第二大元素师两两比较中第二大的元素，再找到第二大元素需要logn-1次比较，加在一起得到如上结果。

错。应该是max，O代表的是算法的上界。

1. 若，，则

对：相当于小于等于。

1. 若，则

对：O相当于大于等于。

1. 贪婪技术所做的每一步选择所产生的部分解，不一定是可行性的。

错：贪心算法每一步必须满足：可行的（即它必须满足问题的约束）、局部最优、不可撤销。

贪心算法通常包含一个用以寻找局部最优解的迭代过程，在某些实例中这些局部最优解转变成了全局最优解，而在另外一些实例中则无法找到全局最优解。

1. Las Vegas算法只要给出解就是正确的。

对。

1. 一个完全多项式近似方案是一个近似方案，其中每一个算法在输入实例的规模的多项式时间内运行。

错：题目中定义的是多项式近似方案。

完全多项式近似方案：是一个近似方案，其中每一个算法在输入实例的规模和两者的多项式时间内运行

1. 问答题
2. **二叉查找树属于减治策略的三个变种中的哪一个的应用？什么情况下二叉查找树表现出最差的效率？此时的查找和插入算法的复杂性如何？**
3. 二叉查找树属于减治策略中减少可变规模变种中的应用；
4. 当树的高度=节点个数-1，即树是一颗严格歪树时表现出最差的效率；
5. 此时查找和插入算法的时间复杂度为O(n)。

减治法的三种变种：

1. 减去一个常量：

递归法求;

插入排序；最坏 平均

快速排序+插入排序

拓扑排序：减一

生成排序：减一 要求1-n的排序先求1-（n-1）的排序

生成子集：减一

1. 减去一个常数因子：

二分搜索

假币问题

俄式乘法

约瑟夫问题

1. 减去可变规模

欧几里得除法

插值查找：最差O(n)

二叉查找树:最差O(n) 平均O(logn)

变治的3种类型：

1. 实例化简：变换为同样问题的一个更简单或更方便的实例

预排序：如果列表是有序的，许多问题更容易求解。

-检验数组中元素的唯一性：先对数组排序，然后只检查连续元素，如果该数组有相等的元素则一定有一对元素是相互紧挨着的 复杂度：排序+查找紧邻元素是否相同即O（nlogn+n）=o(nlogn)。

-查找数字列表中出现频率最高的一个数：先对输入排序（之后有相等的数值都会邻接在一起），求出该有序数组中邻接次数最多的等值元素。

高斯消去：把n个线性方程构成的n元联立方程组变换为一个等价的方程组，该方程组有一个上三角系数矩阵，该矩阵主对角线下的元素都为0.

LU分解

AVL树：把集合变为二叉树

-AVL树是一棵二叉树，其中每个节点的平衡银子定义为该节点左子树和右子树的高度差，这个平衡因子要么是0要么是1要么是-1.

1. 改变表现：变换为同样实例的不同表现

2-3树：平衡查找树

堆和堆排序

霍纳法则

二进制幂

1. 问题化简：变换为另一个问题的实例，这种问题的算法是已知的

最小公倍数：通过两数乘积与其最大公约数之商求得

函数极值：已知最小求最大，目标函数加负号

线性规划

简化为图

1. **何谓伪多项式算法？如何将一Monte Carlo算法转化为Las Vegas算法？**
2. 伪多项式算法是一种在L值的多项式时间内运行的算法，其中L是输入实例中的最大数值；
3. Monte Carlo算法每次都能得到问题的解，但不保证所得的解的准确性；Las Vegas算法每一次不一定得到问题的解，只要得到解一定是正确的解。可以在Monte Carlo算法后加上一个验证算法，如果正确就得到解，如果错误就不能生成问题的解，这样Monte Carlo算法便转化成了Las Vegas算法。

可以通过多次运行Monte Carlo，并且满足每次运行时的随机选择都相互独立，使产生非正确解的概率可以减到任意小，从而转化为Las Vegas。

1. **构造AVL树和2-3数的主要目的是什么？它们各自有什么样的查找和插入的效率？**
2. 构造AVL树和2-3树的目的是使左右子树更加平衡，减小树的层数，使平均搜索效率更高；

（避免出现2叉树中，树严格歪斜的情况，使以每个节点为根的左右子树的高度接近，增加查找和插入的效率。）

1. 他们在最坏情况下插入和查找的时间复杂度为O(log2n)。
2. **写出0/1背包问题的一个多项式等价(Polynomial Equivalent)的判定问题，并说明为什么它们是多项式等价的。**

（1）0/1背包问题的一个多项式等价判定问题是：给定价值为V1,V2,..,Vn,重量为w1,w2,..,wn的n个项和两个整数v\*和w\*，是否存在一个子集S使得且。

（2）若存在0/1背包问题的多项式算法，则可用其在多项式时间内求解判定问题，令背包容量等于w\*，求出0/1背包问题的最优子集S，则可以通过判断S是否满足来确定判定问题的解。

若存在判定问题的多项式算法，则可以在可能的价值范围内进行二分搜索，在各搜索点上解判定问题以确定0/1背包问题的最优解，令，可在O(logV)时间内求解。

1. **下面问题是否属于NP问题？为什么？**

问题表述：给定图中的两个点、，整数和，图中每条边的长度及便利这条边的时间，问图中是否存在一条由到的路径，使得其长度大于，且遍历时间小于？

属于NP问题，因为给定一个肯定实例能够在多项式时间O(n)内验证。

1. 算法题
2. 写出一个求解下列问题的分治算法，推导其时间复杂性并与蛮力法相比较。

给定互不相等的n个数的一个序列，若其中某两个数和的关系为：且，则称和是逆序的。要求计算该序列中的逆序个数，即具有逆序关系的元素对的总数目。

（类比合并排序的分治算法）

1. 算法思想：将数组对半分成两个子数组，分别计算两个子数组中的逆序个数，再计算两个子数组之间的逆序数。

伪代码：

findReverse(A,low,high)

{

Num=0;

if(low<high-1)

{mid=(low+high)/2;

findReverse(A,low,mid);

findReverse(A,mid+1,high);

Reverse(A,low,mid,high);

}

else

{ if(A[low]>A[high-1])

Num=num+1;

}

}

Reverse(A,low,mid,high)

{

I=low;

J=mid+1;

for(i=low;i<=mid;i++)

{

If(A[i]>A[j])

Num=num+1;

}

}

1. 时间复杂度的递推公式为（n=1时，函数应该为0），则使用分治策略后的算法时间复杂度为O(nlogn).
2. 蛮力算法下每个元素都要与数组中它之后的元素比较确定逆序对数，蛮力算法的时间复杂度为O(n2)。当n较大时，分治法计算规模小于蛮力算法。
3. **为一个整数序列，中的整数如果在中出现次数多余，那么称为多数元素。例如，在序列中3是多数元素，因为出现了4次，大于。求A的多数元素问题的蛮力算法复杂性如何？设计一个具有变治思想的算法，提高蛮力算法的效率，写出伪代码并分析其事件复杂性。**

（类比实例化简中的模式运算、预排序）

1. **蛮力算法**：对数组进行扫描，在另外两个数组中存储已经遇到的值和它的频率，每次迭代时，通过遍历储值辅助数组，使原始数组中第i个元素和已遇到的数值进行比较，如果碰到一个匹配值，该数值出现的频率加1，否则，将当前元素添加到储值数组中，并把它出现的频率置为1.最后检查频率数组中是否有元素大于，若有则对应的储值数组中的元素为多数元素。蛮力算法的时间复杂度为O(n2)。
2. **变治思想的算法**：先将数组中的元素按非递减顺序排序，检查该有序数组中邻接次数最多的等值元素个数是否大于。

伪代码：

findMostNum(A[1…n])

{

对数组A排序；

I=1;

While(i<=n)

{

Length=n/2;

While(i+length<=n&&A[i]=A[i+length])

Length=length+1;

If(length>=n/2)

输出A[i];

I=i+length;

}

}

该算法的时间复杂度为排序的时间复杂度和线性时间复杂度之和即O(nlogn)。

1. **动态规划**
2. **某工厂调查了解市场情况，估计在今后四个月内，市场对其产品的需求量如下表所示。**

|  |  |
| --- | --- |
| 时期（月） | 需要量（产品单位） |
| 1 | 2 |
| 2 | 3 |
| 3 | 2 |
| 4 | 4 |

已知：对每个月来讲，生产一批产品的固定成本费为3（千元），若不生成，则为零。每生产单位产品的成本费为1（千元）。同时，在任何一个月内，生产能力所允许的最大生产批量为不超过6个单位。

又知：每单位产品的库存费用为每月0.5（千元），同时要求在第一个月开始之初，及在第四个月末，均无产品库存。

问：在满足上述条件下，该厂应如何安排各个时期的生产与库存，使所花的总成本费用最低？写出你所设的状态变量、决策变量、状态转移方程与递推关系式，和手工求解的详细步骤及结果。

1. **用动态规划方法手工求解以下问题**

有8万元的投资可以投给3个过目，每个项目在不同筒子数额下（以万元为单位）的利润如下表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 投资额  盈利  项目 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 项目1 | 0 | 5 | 15 | 40 | 80 | 90 | 95 | 98 | 100 |
| 项目2 | 0 | 5 | 15 | 40 | 60 | 70 | 73 | 74 | 75 |
| 项目3 | 0 | 4 | 26 | 40 | 45 | 50 | 51 | 52 | 53 |

请安排投资计划，使总的利润最大。

写出你所设的状态变量、决策变量、状态转移方程与递推关系式和手工求解的详细步骤及结构。

1. 分支定界
2. 用分支定界法求解以下问题：

某部门欲建立联通分布于五个区的共50个站点的有线通讯网络，每两个站点之间的线路敷设费用由对成矩阵C给出，任意两站点之间敷设线路需建设的地井数目由对称矩阵U给出。

设计一线路敷设总费用为最小的无环网络，使得徐建设的总地井数目不超过UMAX，且需跨区敷设的线路总数目不超过DMAX(个站点所属的区由向量D给出)。

1. 说明你是如何构造搜索树的。（要求是二叉搜索树）
2. 说明算法遍历搜索树的原则。（何时以及如何前进、分支、回溯、剪枝等等）
3. 你设计的分支定界算法的“界”是什么，他为什么是正确的和有效的？
4. 写出伪代码。

解：

1. 构造一棵二叉搜索树，根节点表示起始站点，叶节点表示其余站点，每层的分支对应于是否把某条边添加到解集中的决策。沿左分支前进表示选择该边，沿右分支前进表示不选该边。例如，若ai左子树为aj，则表示选择边aij到解集中，若ai的右子树为aj，表示不添加aij到解集。
2. 前进：总是从树的根节点开始，当线路费用小于DMAX并且地井数目小于UMAX且图中无环时前进，当下行时总是先沿着左分支进行。

分支：当一个节点下一步有多个选择时分支。

回溯：当有如下情况之一时进行回溯：

1. 当前挑选的边使得解不可行；
2. 已经找到一个解

当从左分支回溯到某顶点时，接着沿其右分支向下进行；

当从右分支回溯到某顶点时，接着回溯到其父顶点。

剪枝：当前路径不满足可行性要求或超出界时。

1. 题目要求设计一个无环网络使得线路总数不超过DMAX、地井总数不超过UMAX。所以问题初始时的界为：线路总数=DMAX且地井总数=UMAX。

当扩展出的节点满足：1.线路无环；2.线路总数不大于DMAX；3.地井总数不大于UMAX；4.线路覆盖所有站点时，更新问题的上届，此时获得了一个比原来更紧的界。

1. 用分支定界法求解以下问题：

A国与B国之间尚未直接通商。与A国直接通商的有20个国家（C1, C2, …, C20）；与B国直接通商的为另外30个国家（C21, C22, …, C50）。上述50个国家之间并不是每两个国家都直接通商，任意两国之间的贸易税率由对称矩阵R给出，其中代表两国不能直接通商。

A国某公司与B国一公司欲通过某几个中间国家的公司完成一笔贸易，各个国家的进出口贸易通关等手续所需办理时间由向量T给出。请安排一中转贸易计划，使得该交易所产生的向各中转国缴纳的税费最低，且整个交易能够在时间t内完成。

1. 说明你是如何构造搜索树的。（要求是二叉搜索树）
2. 说明算法遍历搜索树的原则。（何时以及如何前进、分支、回溯、剪枝等等）
3. 你设计的分支定界算法的“界”是什么，他为什么是正确的和有效的？
4. 写出伪代码。

解：

1. 构造二叉搜索树，根节点为A国，其余节点代表剩余国家。每层分支对应于是否在两国之间中转贸易。沿左分支前进表示在两国之间中转贸易，沿右分支前进表示不在两国之间中转贸易。例如，若ai的左子树为aj，则表示在ai和aj两国之间中转贸易；若ai的右子树为aj则表示不在ai和aj两国之间中转贸易。
2. 前进：总是从树的根节点开始，当交易税费小于目前得到解的税费值且办理手续时间小于T时前进，当下行时总是先沿着左分支前进。

分支：当一个节点下一步有多个选择时分支。

回溯：当满足下列情况之一时回溯：

1. 当前挑选的边使得解不可行；
2. 已经找到一个解。

当从左分支回溯到某顶点时，接着沿其右分支向下进行；

当从右分支回溯到某顶点时，接着回溯到其父顶点。

剪枝：当前路径不满足可行性要求或超出界时。

（3）问题的上界是目前已经得到的一个可行解的总税费，当产生更好的界的时候更新界。