**一：判断题**

1. 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）
2. NP完全问题比其他所有NP问题都要难。 （错）

NPC问题至少与其他所有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）

使用深度优先法。

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就构成一个决策序列，通常称为一个策略。

1. P类和NP类问题的关系用P⊂NP来表示是错误的。 （错）

P⊂NP成立

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

F(s\_a)/F(s\_a/3)

1. 通常来说，算法的最坏情况的时间复杂度比平均情况的时间复杂性容易计算。 （对）
2. 若P2多项式时间转化为(polynomial transforms to) P1，则P2至少与P1一样难。 （错）

解出 P1 就解出了 P2 。解P1至少与解 P2一样难，反之不一定。

1. 快速排序算法的平均时间复杂度是O(n log⁡n)，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）

随机化快速排序只是降低了最坏时间复杂度。

1. 基于比较的寻找数组A[1,…,n]中最大元素的问题下届是Ω(n/3)。 （对）

但不是最优下界

1. O(f(n))+O(g(n))=O(min⁡〖{f(n),g(n)})〗 （错）

Max

1. 若f(n)=Ω(g(n))，g(n)=Ω(h(n))，则f(n)=Ω(h(n)) （对）
2. 若f(n)=O(g(n))，则g(n)=Ω(f(n)) （对）
3. 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）

局部最优的可行解

1. Las Vegas算法只要给出解就是正确的。（对）
2. 一个完全多项式近似方案是一个近似方案{A\_ε}，其中每一个算法A\_ε在输入实例I的规模的多项式时间内运行。 （错）

多项式近似。

**二：简答**

1. 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？

答：减治策略有3个主要的变种，包括减常量、减常数因子和减可变规模。(1) 二叉查找树属于减可变规模变种的应用。(2) 当先后插入的关键字有序时，构成的二叉查找树蜕变为单支树，树的深度等于n，此时二叉查找树表现出最差的效率，(3) 查找和插入算法的时间效率都属于Θ(n)。

1. 何谓伪多项式算法？如何将一Monte Carlo算法转化为Las Vegas算法？

答：伪多项式时间算法是一种算法，它在L值的多项式时间内运行，其中L是输入实例中的最大数值。

Las Vegas算法不会得到不正确的解。一旦用拉斯维加斯算法找到一个解，这个解就一定是正确解。但有时用拉斯维加斯算法找不到解。

Monte Carlo算法每次都能得到问题的解，但不保证所得解的准确性

转化：可以在Monte Carlo算法给出的解上加一个验证算法，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算法便转化为了Las Vegas算法。

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？

答：(1)当先后插入的关键字有序时，构成的二叉查找树蜕变为单支树，树的深度等于n，此时二叉查找树表现出最差的效率，为了解决这一问题，可以构造AVL树或2-3树，使树的深度减小。一棵AVL树要求它的每个节点的左右子树的高度差不能超过1。2-3树和2-3-4树允许一棵查找树的单个节点不止包含一个元素。(2) AVL树在最差情况下，查找和插入操作的效率属于Θ(lgn)。2-3树无论在最差还是平均情况下，查找和插入的效率都属于Θ(log n)。

1. 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。

答：0/1背包问题：从M件物品中，取出若干件放在空间为W的背包里，给出一个能获得最大价值的方案。每件物品的体积为W1，W2……Wn，与之相对应的价值为P1,P2……Pn。

判定问题I：从M件物品中，取出若干件放在空间为W的背包里，是否存在一个方案，所获价值≥P\*？。每件物品的体积为W1，W2……Wn，与之相对应的价值为P1,P2……Pn。

若判定问题I存在多项式时间的解法，则反复调用该算法就可以在多项式时间内解决0/1背包的优化问题。因而这个判定问题与原问题多项式等价。

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？

答：这个问题属于NP问题。因为若给出该问题的一个解，可以在多项式时间内检验这个解的正确性。如给出一条由p到q的路径，可以在多项式时间内计算出它的长度及遍历时间，然后分别与C和t进行比较，从而可以判断这个解的对错。

**三：分治**

1. 多数元素

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。

蛮力算法时间复杂度：O(n2)；空间复杂度：O(1)

/\*\*

\*对长度为len的数组A，找出它的多数元素

\*/

majority(A[], len) {

HeapSort(A); // 对数组A中元素按从小到大顺序进行堆排序

i = 0;

flag = false;

while i < len {

tmp = A[i]

count = 1;

while i + count < len && A[i+count] == tmp {

count++;

}

if count > len/2 {

print(tmp+“是多数元素”)

flag = true;

break;

}

i + = count;

}

if flag == false

print(“没有多数元素”)

}

该算法的时间复杂性：O(nlgn)

1. 逆序数

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a\_1,a\_2,…,a\_n，若其中某两个数a\_i和a\_j的关系为：a\_i>a\_j且i<j，则称a\_i和a\_j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。

/\*\*

\*求解n个数的一个序列，具有逆序关系的元素对的总数目

\*/

count = 0; //逆序元素对的全局计数变量

mergeInvertedPairs(A,low,mid,high) {

i = low;

j = mid+1;

k = low;

tmp[n]; //用于归并排序的辅助数组

while i <= mid && j <= high {

if (A[i] > A[j]) {

tmp[k] = A[j++];

count += (mid-i+1); //相比归并排序，就多了这一条语句

} else {

tmp[k] = A[i++];

}

k++;

}

while i <= mid {

tmp[k++] = A[i++];

}

while j <= high {

tmp[k++] = A[j++];

}

for (j = low; j <= high; j++) {

A[j] = tmp[j];

}

}

findInvertedPairs(A[], low, high) {

if (low < high) {

mid = (low+high) / 2;

findInvertedPairs(A,low,mid);

findInvertedPairs(A,mid+1,high);

mergeInvertedPairs(A,low,mid,high);

}

}

**四、动态规划**

**2011年**

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。

**解：**

状态变量：xk 表示留给项目k..n的投资额，其中n为项目总个数，k=1..n.

决策变量：uk 表示投给项目k的投资额.

允许决策集合：

状态转移方程：

递推关系式：

其中，表示项目k的投资额为uk时的盈利.

针对本题，n = 3，xk最大取8

手工详解过程：

1. **初始化k = 3**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **x3** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **f3(x3)** | 0 | 4 | 26 | 40 | 45 | 50 | 51 | 52 | 53 |

1. **k = 2**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **x2** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **f2(x2)** | 0 | 5 | 26 | 40 | 60 | 70 | 86 | 100 | 110 |

1. **k = 1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **x1** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **f1(x1)** | 0 | 5 | 26 | 40 | 80 | 90 | 106 | 120 | **140** |

**最终结果：**给项目1投资4万元，项目2投资4万元，项目3不投资，将获得最大利润140万元.

2012年

1. 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。

**解：**

设阶段序数k表示月份，状态变量xk为第k个月初拥有的单位产品数量，亦为第k-1月末时的单位产品数量，决策变量uk为第k个月生产的单位产品数量，ck为第k月份需要的产品数量，这里xk和uk均取离散变量。

状态转移方程为：

, k =1,2,3,4; 且x1=0。

k段允许决策集合为:

k = 1,2,3;

当k=4时，。

设为第k月的成本费,单位为(千元)，则

,

故指标函数为

令表示为由出发采用最优生产方案到第4个月结束这段期间的产品成本。

根据最优化原理，则有递推公式：

其中：

逆序计算的详细步骤如下：

1. 当k=4时，
2. 当k=3时，因为 且，所以有：

当处取得最小值。

当,此时处取得最小值。

当,此时处取得最小值。

当,此时,在处取得最小值。

当,此时,在处取得最小值。

当,此时,在处取得最小值。

当,此时,在处取得最小值。

1. 当k=2时，因为，且所以有：

当时，,在处取得最小值。

当时，,且在处取得最小值。

当时，,在处取得最小值。

当时，，且在处取得最小值。

当时，,且在处取得最小值。

当时，,且在处取得最小值。

当时，,且在处取得最小值。

1. 当k=1时，因为,所以有：

当,,且在处取得最小值。

综上所述，最优的库存方案为：第一月生产5单位产品，第二月和第四月不生产，第三月生产6单位产品。

五：分支限界

1. 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。
5. 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。

**别人答案1：**

答：1.（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.当需跨区敷设的线路总数目小于DMAX，地井数目小于UMAX时前进

分支：当一个节点的下一步有多个选择时，进行分支

回溯：当前路径不满足要求时，回溯。如果是从左孩子回溯，则搜素父节点的右孩子；如果从右孩子回溯，则接着回溯到其父顶点。

剪纸：当前路径不满足要求，或者超出界时

3.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。

4. 分支（线路e，是左子树还是右子树）

begin

if (是左子树)

begin

add 线路e to 选择的线路集合；

if（选择的线路集合中线路条数＞49 or 选择的线路集合会形成环 or 当前敷设费用＞最小费用 or 当前总地井数＞UMAX or 当前跨区线路数＞DMAX） return；

if （选择的线路集合中线路条数为49 && 这些线路是联通的）

begin

最小费用 = 当前敷设费用；

最优方案 = 选择的线路集合；

return；

end；

end；

线路next = 线路e的下一条未访问的线路；

线路next访问标记 = 已访问；

//遍历左子树

当前敷设费用 += 线路next的敷设费用；

当前总地井数 += 敷设线路next需要建设的地井数；

if（线路next是跨区线路） 当前跨区线路数++；

分支（线路next，左子树）；

当前敷设费用 -= 线路next的敷设费用；

当前总地井数 -= 敷设线路next需要建设的地井数；

if（线路next是跨区线路） 当前跨区线路数--；

//遍历右子树

分支（线路next， 右子树）；

线路next访问标记 = 未访问；

end；

**别人的答案2：**

1. 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。

2. 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。

**其他答案：**

1.

1. 说明你是如何构造搜索树的。（要求是二叉搜索树）

50个站点之间最多有50\*49/2条边，对这些边进行编号，则每条边表示二叉树的一个节点，当节点k选左孩子时则表示选择了该边，为右孩子时，则表示不选择序号为k的边，二叉树最坏有50\*49/2层

1. 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）
2. 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？

UMAX DMAX 当前最短路径的费用 无环

1. 写出伪代码。

void LL()

{

int feeLowBound=INF;//初始化最小费用的界

A[]=init();//初始化矩阵C，D，为边标序号,A为边集合

Tree=buildtree(A);//以A建立二叉搜索树

while(not over) //循环到遍历结束

{

if( //符合以下的界

countminU()<UMAX //计算当前路径满足条件的最小数目

&& countminD()<DMAX //计算当前路径满足条件的最小跨区数目

&& countminfee() <feeLowBound //计算当前路径的最小花费

&& hasnocircle() //没有出现环

)

{

if(findscheme) //找到一个方案，则这个方案必然小于lowbound

{

feeLowBound=countminfee();

record(route);//更新界，记录路径

backward();//回溯

}

else //前进

{

movetonextchild();

continue;

}

}

else {

backward();//回溯&剪枝

}

}

}

2.

1. 说明你是如何构造搜索树的。（要求是二叉搜索树）

因为解空间每个解容量为50，存在250种解的情况，故需要构建的二叉搜索树的层数为50，父节点ai的左孩子结点a2i-1表示选择该结点，右孩子a2i+1表示不选择该结点。

1. 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）

前进：线路费用小于DMAX，地井数目小于UMAX；

分支：一个节点的下一步有多个可行的选择时进行分支；

回溯：当前路径归纳当前节点后不满足T限制时进行回溯，如果当前节点是左孩子结点，则回溯到其兄弟结点的右孩子结点；如果当前结点是右孩子结点则回溯到其祖先结点的右孩子结点；

减枝：当不满足T的限制时对该节点所在的进行减枝，出现环时进行减枝；税费大于当前可行路径

1. 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？

界：T，当前的最小税费；

1. 写出伪代码。