**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）

3、回溯法用深度优先法或广度优先法搜索状态空间树。（错，仅深度优先）

4、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）

5、P类和NP类问题的关系用P⊂NP来表示是错误的。（错）

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

7、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）

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

9、快速排序算法的平均时间复杂度是O(nlogn)，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）

10、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)。（错）

11、O(f(n))+O(g(n))=O(min{f(n),g(n)})（错）

12、若f(n)=Ω(g(n))，g(n)=Ω(h(n))，则f(n)=Ω(h(n))（对）

13、若f(n)=O(g(n))，则g(n)=Ω(f(n))（对）

14、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）

15、LasVegas算法只要给出解就是正确的。（对）

16、一个完全多项式近似方案是一个近似方案{Aε}，其中每一个算法Aε在输入实例I的规模的多项式时间内运行。（错）

**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：若一个数值算法的时间复杂度可以表示为输入数值N的多项式，但其运行时间与输入数值N的二进制位数呈指数增长关系，则称其时间复杂度为伪多项式时间。

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树的主要目的是什么？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。

分治题

1. 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。

**解：**/\*\* \*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);

}

算法思路：以归并排序为基础，在两两集合合并的时候如果前一个集合的元素a[i]>a[j]，那么说明需要调整次序，逆序数num=num+mid-i。

时间复杂度的迭代公式为 因此算法的时间复杂度为T(n)=O(nlogn)；

蛮力法的时间复杂度为O(n2)，当n数目较大时，分治法计算规模远小于蛮力法。

1. 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。

2. num **<-** src**[**0**];**

count **<-** 0**;**

for i **<-** 0 to n**-**1 do

**{**

if**(**num **==** src**[**i**])**

**{**

count**++;**

**}**

else

**{**

count**--;**

if**(**count **<** 0**)**

**{**

num **<-** src**[**i**];**

**count = 0;**

**}**

**}**

**}**

采用减治的思想每一个减去一个元素，时间复杂度为O(n)，蛮力法的时间复杂度为O(n2)。

动态规划题

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。

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 |

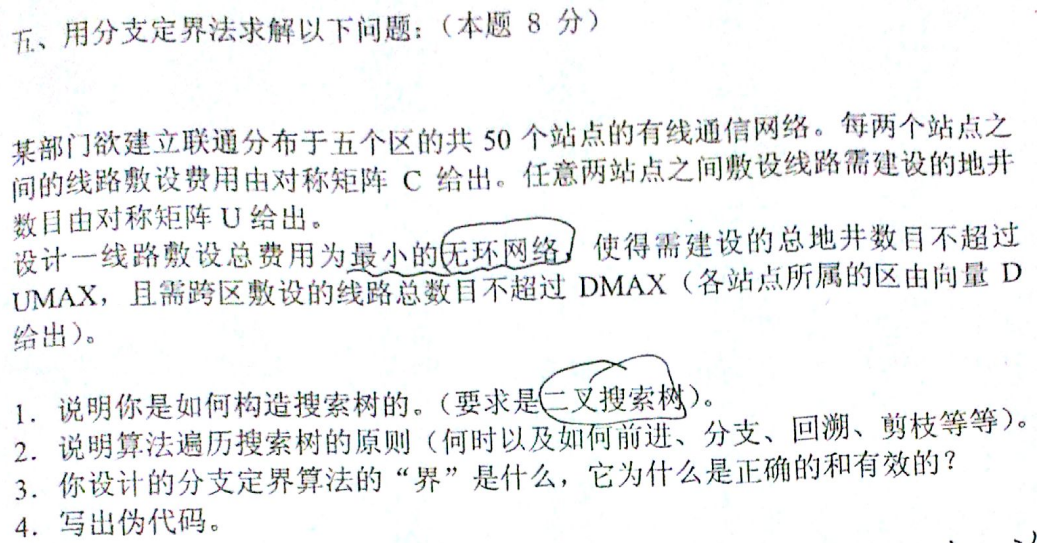
1. **k = 2**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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.



线路题的某种深搜解法：

1)可以根据线路(l1,l2,...,lm)的取舍构建一棵m层二叉搜索树。第i层的所有左分支表示铺设线路li，右分支则表示不铺设。如果存在可行解，遍历此二叉搜索树即可找到最优解。

2)

前进：当前节点未被剪枝并且仍有子节点即可继续前进。

分支：先遍历左分支，后遍历右分支。

回溯：左右分支都被遍历时返回父节点。

剪枝：

剪枝条件如下：

1。有环路

2。当前地井数 + 地井数下界 > UMAX

3。当前跨区铺设线路数 + 跨区铺设线路数下界 > DMAX

4。当前费用 + 费用下界 >= 已知最优方案的费用

3)

子问题的下界为费用下界、地井数下界、跨区线路数下界。费用下界是根据剩余站点数量定义的，累计最小的路线花费即可得到。由于限制被极度弱化，所以非常粗糙，但是正确有效。另外两个下界也类似。父问题的上界是已知最优方案的费用，显然正确有效。

4)

按费用从小到大排序所有路线l1,l2,...,lm

计算子问题下界：

1。费用下界：剩余站点数量->最小花费 #累计最小的线路花费即可得到，下同

2。地井数下界：剩余站点数量->最小地井数

3。跨区线路数下界：剩余站点数量->最小跨区线路数

search(空集, l1)

返回最优结果

def search(线路集合S，当前线路l):

判断线路集合S是否合格，条件如下：

1。无环路

2。当前地井数 + 地井数下界 <= UMAX

3。当前跨区铺设线路数 + 跨区铺设线路数下界 <= DMAX

4。当前费用 + 费用下界 < 已知最优方案的费用

如果合格：

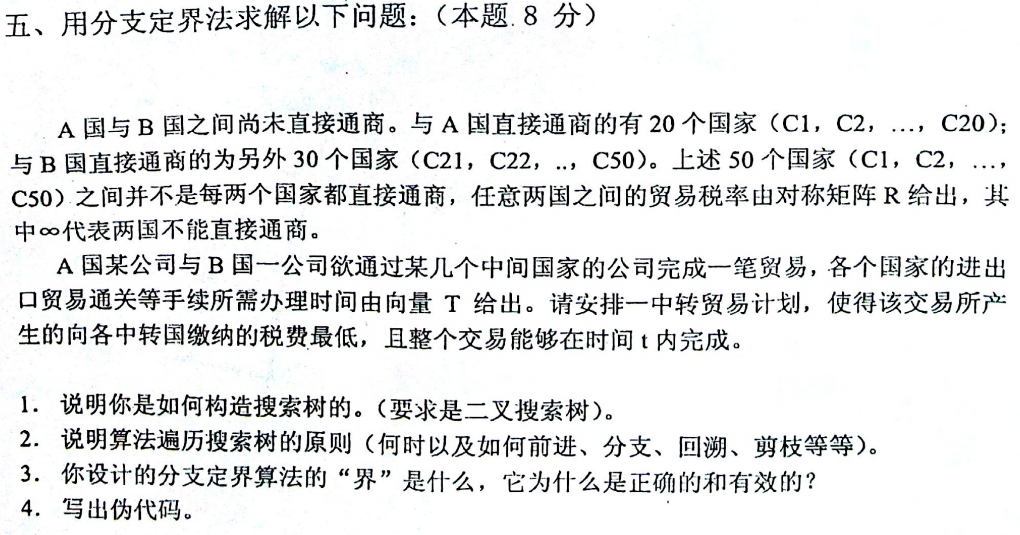
当前网络已经覆盖所有站点:

记S为已知最优

否则若剩下的线路数有可能使所有站点构成网络:

search(S ∪ {l}, l的下一条路线)

search(S, l的下一条路线)



税费题的某种深搜解法：

1)

可以根据除A外的51个国家定义一棵若干层二叉搜索树。每个节点的左分支表示选择其代表

的国家为下一个贸易顺序上的国家，右分支则表示不选择。构造搜索树需要两个辅助变量，

之前的贸易顺序S（s为S的最后一个国家）和这一轮否决的国家V。任取可以和s国贸易的国家c（不属于S和V）置于树的当前生成位置，然后用(S' = <S, c>和V' = 空集)生成左子树，用(S' = S和V' = V ∪ {c})生成右子树。如果c不存在或者s = B则终止当前子树的生成。如此反复可以建立一棵二叉搜索树。

2)

前进：当前节点未被剪枝并且仍有子节点即可继续前进。

分支：先遍历左分支，后遍历右分支。

回溯：左右分支都被遍历时返回父节点。

剪枝：

剪枝条件如下：

1。当前税费 + s国与B国贸易的最小税费 >= 已知最优方案的税费

2。当前时间 + s国与B国贸易的最短时间 > t

3)

子问题的下界为税费下界和时间下界，均由dijkstra算法算法得到，表示某国与B国贸易的最小税费和最短时间。两个结果均由弱化限制的方法得到，所以是正确的，计算复杂度也不高，当然有效。

父问题的上界是已知最优方案的税费，显然正确有效。

4)

使用dijkstra算法得到子问题下界：

1。税费下界：某国与B国贸易的最小税费，顺便记录对应的时间和贸易顺序

2。时间下界：某国与B国贸易的最短时间

search(<A>)

返回最优结果

def search(贸易顺序S):

令s为S的最后一个国家

判断S是否合格，条件如下：

1。当前税费 + s国与B国贸易的最小税费 < 已知最优方案的税费

2。当前时间 + s国与B国贸易的最短时间 <= t

如果合格：

当前时间 + s国与B国贸易的最小税费对应的时间 <= t:

记<S, s国与B国贸易的最小税费对应贸易顺序(不包括s)>为已知最优

否则对所有可以与s国贸易的国家c:

search(<S, c>)