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
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. 类和类问题的关系用来表示是错误的。 错
6. 若近似算法A求解某极小化问题一实例的解为，且已知该问题的最优解为，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，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中最大元素的问题下届是。 对
11. 错
12. 若，，则 对
13. 若，则 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. Las Vegas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，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

问答题

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？

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. 何谓为多项式算法？如何将一Monte Carlo算法转化为Las Vegas算法？

伪多项式算法：算法的时间复杂度是输入数据的多项表达式，但确实输入长度的指数时间算法

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；Las Vegas算法是每次不一定得到问题的解，只要得到的解一定是正确的解；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？

构建AVL树和2-3树能够平衡左右子树的高度，减少树的层数，使得平均搜索效率更高。

插入和查找的时间复杂度均为O(logn)。

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。

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？

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。

int count=0;

void merge(int arr[],int first,int mid,int last)

{

int tmpArr[100];

int i=first,j=mid+1;

int cur=0;

while(i<=mid && j<=last)

{

if(arr[i]<arr[j])

{

tmpArr[cur++]=arr[i++];

}

else

{

tmpArr[cur++]=arr[j++];

count += mid-i+1;//只增加这一句便可求逆序数

}

}

if(i<=mid)

{

while(i<=mid)

tmpArr[cur++]=arr[i++];

}

else

{

while(j<=last)

tmpArr[cur++]=arr[j++];

}

for(int k=0;k<cur;k++)

{

arr[first++]=tmpArr[k];

}

}

void mergeSort(int arr[],int first,int last)

{

if(first==last)

return;

int mid=(first+last)/2;

mergeSort(arr,first,mid);

mergeSort(arr,mid+1,last);

merge(arr,first,mid,last);

}

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。

三种主要类型：实例化建、改变表现、问题化简

#include<stdio.h>

int main(){

int a[] = {1,3,2,3,3,4,3,4,3};

int n = 9;

int seed = a[0];

int count = 1;

for(int i = 1;i < n;i++)

{

if(seed==a[i])

{

count ++;

}else{

if(count>0)

count--;

else

seed = a[i];

}

}

count = 0;

for(i=0;i<n;i++)

{

if(a[i]==seed)

count++;

}

if(count>n/2)

printf("%d\n%d\n",seed,count);

}

在原序列中去除两个不同的元素后，那么在原序列中的多数元素在新序列中还是多数元素。

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。

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。

分支定界题

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）

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

}

}

}

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）

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。