**判断题：**

1-5: T F F T F 6-10: F T F F F 11-16: F T T T T F

**问答题：**

1.（1）属于减可变规模的应用。（2）当关键字的个数等于二叉查找树的高度时表现出最差的效率。时间复杂度为O(n)。（3）此时查找和插入算法在最坏情况（查找当前序列的最大值或者插入最大值）的时间复杂度都是O(n)。

2.（1）伪多项式时间算法是NPC的一种，存在复杂度是关于实例规模和实例所有参数中绝对值最大数是成多项式关系的算法。

（2）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。

3. 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)。

4. 0/1背包问题的一个多项式等价判定问题是整数划分问题。

5. 属于NP问题。因为可以在多项式的时间验证一个候选路径是否符合条件。

**分治题：**

1. // num : 逆序数

num <- 0;

Merge(Type a[], Type left, Type mid, Type right)

{

i <- left, j <- right;

while( i < mid && j < right)

{

if(a[j]) > a[i])

{

num += mid - i;

}

}

}

MergeSort(Type a[], Type left, Type right)

{

if(left < right)

{

mid <- (left + right) / 2;

MergeSort(a, left, i);

MergeSort(a, mid + 1, right);

Merge(a, left, i, right);

}

}

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。

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];

}

}

}

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.

k=1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| month=1 | 2 | 3 | 4 | 5 | 6 |
| cost | 5 | 6.5 | 8 | 9.5 | 11 |
| reservation | 0 | 1 | 2 | 3 | 4 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| k=2 |  |  |  |  |  |  |  |
|  | 2 | 3 | 4 | 5 | 6 |  |  |
| 0 |  |  |  | **9.5** | **11.5** |  |  |
| 1 |  |  | 12 | 14 | 17.5 |  |  |
| 2 |  | 11.5 | 13.5 | 15.5 | 19 |  |  |
| 3 | 11 | 13 | 15 | 17 | 20.5 |  |  |
| 4 | 12.5 | 14.5 | 16.5 | 18.5 | 22 |  |  |
| 5 | **14** | **16** | 18 | 20 | 23.5 |  |  |
| 6 | 16.5 | **17.5** | **19.5** | **21.5** |  |  |  |
| curproduction | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| curmincost | 9.5 | 11.5 | 14 | 16 | 17.5 | 19.5 | 21.5 |
| reservation | 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| month=3 |  |  |  |  |  |  |  |
|  | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 |  |  | **14** | 16.5 | 18.5 | 21 | 23.5 |
| 1 |  | 15.5 | 18.5 | 21 | 23 | 25.5 |  |
| 2 | 14.5 | 17 | 20 | 22.5 | 24.5 |  |  |
| 3 | **16** | 18.5 | 21.5 | 24 |  |  |  |
| 4 | **17.5** | 20 | 23 |  |  |  |  |
| 5 | **19** | 21.5 |  |  |  |  |  |
| 6 | **20.5** |  |  |  |  |  |  |
| curproduction | 7 | 8 | 9 | 10 | 11 |  |  |
| curmincost | 14 | 16 | 17.5 | 19 | 20.5 |  |  |
| reservation | 0 | 1 | 2 | 3 | 4 |  |  |

month=4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 7 | 8 | 9 | 10 | 11 |
| 0 |  |  |  |  | **20.5** |
| 1 |  |  |  | 23 |  |
| 2 |  |  | 22.5 |  |  |
| 3 |  | 22 |  |  |  |
| 4 | 21 |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |

故最小生产投入为20.5（千元），其中投入策略如下表格：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| month | 1 | 2 | 3 | 4 |
| input | 5 | 0 | 6 | 0 |

2.

K：项目的编号，k=1,2,3；

Xk：投资的数目，显然xk=0,1,2,3,4,5,6,7,8；

gk(xk)：把数目为xk钱投资到项目k得到的收益；

fk(xk)：把数目为xk的钱投资得到的最大收益（可以投资的范围为项目k~n）；

状态变量xk: 表示投资的数目；

决策变量uk: 表示投给项目k的投资额；

状态转移方程：fk(xk)=max{gk(uk)+fk+1(xk-uk)}, uk=0,1,……,xk;

fn(xn)= gn(xn)

项目个数n=3，xk最大取8，问题划分为三个阶段：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| k=3 |  |  |  |  |  |  |  |  |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 0 | 4 | 26 | 40 | 45 | 50 | 51 | 52 | 53 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| k=2 |  |  |  |  |  |  |  |  |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | **0** | 4 | **26** | **40** | 45 | 50 | 51 | 52 | 53 |
| 1 | **5** | 9 | 31 | 45 | 50 | 55 | 56 | 57 |  |
| 2 | 15 | 24 | 41 | 55 | 60 | 65 | 66 |  |  |
| 3 | **40** | 44 | 66 | 80 | 85 | 90 |  |  |  |
| 4 | **60** | 64 | **86** | **100** | 105 |  |  |  |  |
| 5 | **70** | 74 | 96 | **110** |  |  |  |  |  |
| 6 | 73 | 77 | 99 |  |  |  |  |  |  |
| 7 | 74 | 78 |  |  |  |  |  |  |  |
| 8 | 75 |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| k=1 |  |  |  |  |  |  |  |  |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|  | 0 | 5 | 26 | 40 | 60 | 70 | 86 | 100 | 110 |
| 1 | 5 |  |  |  |  |  |  | 105 |  |
| 2 | 15 |  |  |  |  |  | 101 |  |  |
| 3 | 40 |  |  |  |  | 110 |  |  |  |
| 4 | 80 |  |  |  | **140** |  |  |  |  |
| 5 | 90 |  |  | 130 |  |  |  |  |  |
| 6 | 95 |  | 121 |  |  |  |  |  |  |
| 7 | 98 | 103 |  |  |  |  |  |  |  |
| 8 | 100 |  |  |  |  |  |  |  |  |

由上表可知：最大获益为140万元，其中对项目1投资4万，项目2投资4万，项目3投资0元。

**分支限界题：**

1.

（1）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。

（2）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DMAX和UMAX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DMAX和UMAX的限制时对该节点所在的进行减枝，出现环时进行减枝；

（3）界：线路费用，地井数目，当前的最小花费；

（4）