判断题：

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

问答题：

1.（1）属于减可变规模变种的应用。（2）当节点个数等于二叉查找树的高度时表现出最差的效率。（3）此时查找和插入算法在最坏情况的时间复杂度都是 O(n)。

2.（1）**伪多项式时间算法是一种在 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。**

（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. **AVL 树和 2-3 树能够维持树的平衡，避免树的退化，它们在最坏情况下插入和查找的时间复杂度均为 $O(\log_2 n)$。**

4. 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 V1, V2, …, Vn 的 n 个项和两个整数 v*和 w*，问是否存在一个子集 S，使得 $\sum_{i\in s} v(i) \ge v^*$，且$\sum_{i\in s} w(i) \le w^*$。

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该判定问题，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 $\sum_{i\in s} v(i) \ge v^*$来确定判定问题的解。

若存在该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 V=$\sum_{i\le n} v(i)$ 可在 O(log(V)) 时间内求得解。

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 $T(n)=\begin{cases}1 & n=1;\\ 2T(n/2)+O(n) & n>1\end{cases}$ 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 $O(n^2)$，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 $O(n^2)$。

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：