1. 函数渐进阶。对于下列各组*f*(*x*)和*g*(*x*)，确定他们的关系（15分）
   1. *f*(*x*)=*log* *n*10+1；*g*(*x*)= *log* *n* – 10
   2. *f*(*x*)=5× *n*10；*g*(*x*)=10*n*
   3. *f*(*x*)=；*g*(*x*)= *log* *n* +5
2. 设n个不同的整数排好序后存于T[0:n-1]中。若存在下标i，0≤i<n，使得T[i]=i，试设计一个时间复杂度为O(logn)的算法找到该下标（15分），并用C/C++编程实现（10分）。
3. 分别用贪心算法（20分）、动态规划法（20分）设计求解0-1背包问题的算法。要求：1.说明算法策略（每种方法5分）；2.主要算法步骤（10分）；3.分析算法的时间复杂度（5分）
4. 试设计解决TSP问题的贪心算法（10分），分析时间复杂度（5分），试分析是否存在O(*n*2*log* *n*)的有效算法（5分）。

1函数渐进阶。对于下列各组*f*(*x*)和*g*(*x*)，确定他们的关系（15分）

* 1. *f*(*x*)=*log* *n*10+1；*g*(*x*)= *log* *n* – 10
  2. *f*(*x*)=5× *n*10；*g*(*x*)=10*n*
  3. *f*(*x*)=；*g*(*x*)= *log* *n* +5



2设n个不同的整数排好序后存于T[0:n-1]中。若下标i，0≤i<n，使得T[i]=i，试设计一个时间复杂度为O(logn)的存在算法找到该下标（15分），并用C/C++编程实现（10分）。

解答：由题可知，数组T[ ]是排列好的整数集。可以用二分搜索的算法对问题进行操作。取中间值T[mid],如果T[mid]=mid则可以返回mid；如果T[mid]<mid在mid+1到n-1之间进行上述操作；如果T[mid]>mid在0到mid-1之间进行上述操作。

Int Findi(int T[],int m,int n)

{

Int mid=(m+n)/2;

If (T[mid]==mid) return mid;

else if(T[mid]>mid) return Findi(T[],m,mid-1);

else return Findi(T[],mid+1,n);

}

算法设计：

输入：一个数组T[ ]，起始地址m，末端地址n

比较指定数组位置的中间元素T[(m+n)/2]与中间地址(m+n)/2.

1）如果T[(m+n)/2]等于(m+n)/2，则返回(m+n)/2；

2）如果T[(m+n)/2]大于(m+n)/2，则返回在数组T[ ]中m位到(m+n)/2-1之间进行此算法的返回值；

3）如果T[(m+n)/2]小于(m+n)/2，则返回在数组T[ ]中(m+n)/2+1到n之间进行此算法的返回值；

输出：一个值，若存在则返回i；若不存在则返回0.

3分别用贪心算法（20分）、动态规划法（20分）设计求解0-1背包问题的算法。要求：1.说明算法策略（每种方法5分）；2.主要算法步骤（10分）；3.分析算法的时间复杂度（5分）

4试设计解决TSP问题的贪心算法（10分），分析时间复杂度O(*n*2)（5分），试分析是否存在O(*n*2*log* *n*)的有效算法（5分）。

答：

**贪心策略**：在当前节点下遍历所有能到达的下一节点，选择距离最近的节点作为下一节点。

**基本思路：**

1. 从一节点出发遍历所有能到达的下一节点，选择距离最近的节点作为下一节点；
2. 然后把当前节点标记已走过，下一节点作为当前节点，重复贪心策略，以此类推直至所有节点都标记为已走节点结束。

3、用最短路径问题寻找到达始发点的最短路径，返回出发点。