**实验三 蛮力法（枚举）**

1. **实验目的**
2. 掌握蛮力算法的基本思想。
3. 熟练掌握蛮力算法求解问题的过程。
4. **实验要求**
5. 能运用蛮力法解决实际问题，加深对蛮力算法的理解及应用；
6. 理解蛮力算法的适用范围，体会蛮力算法的优缺点。
7. 得到正确的程序运行结果
8. **主要仪器及耗材**

计算机及相关软件

1. **实验内容**

1. 设计一个蛮力算法,对于给定的x0,计算下面多项式的值:

P(x)=anxn+an-1xn-1+…+a1x+a0

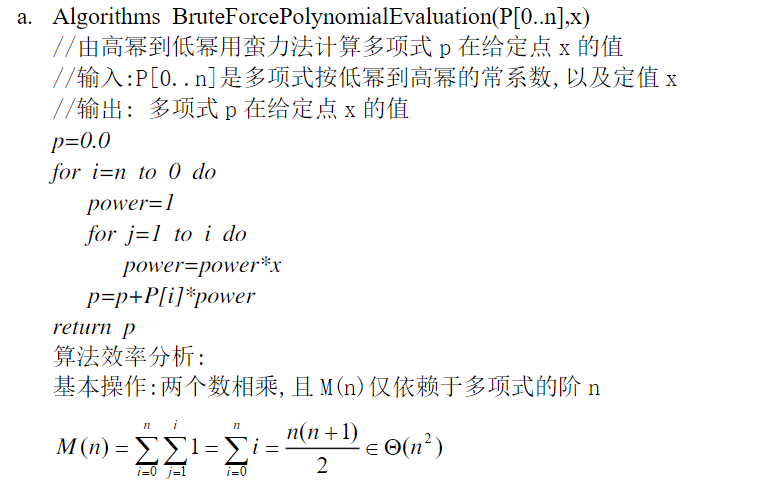
**问题：**

①. 这个算法最坏情况下的效率类型是什么？

②. 如果你设计的算法属于Θ(n2),请你为该算法设计一个线性的算法。

③. 对于该问题来说,能不能设计一个比线性效率还要好的算法呢?

④. 编程实现你设计的算法。



2. 应用选择排序对序列E,X,A,M,P,L,E按照字母顺序排序。算法描述如下：

算法 SelectionSort(A[0..n-1])

//该算法用选择排序对给定的数组排序

//输入：一个可排序数组A[0..n-1]

//输出：升序排序的数组A[0..n-1]

for i←0 to n-2 do

min ←i

for j←i+1 to n-1 do

if A[j]< A[min]

min ←j

swap A[i] and A[min]

**问题：**

①. 选择排序稳定吗？不稳定！

②. 如果用链表实现以上的排序，能不能获得和数组版相同的Θ(n2)的效率？

链表交换，可以的！

③. 编程证明你的结论。

3. 应用冒泡排序对序列E,X,A,M,P,L,E按照字母顺序排序。算法描述如下：

算法 BubbleSort(A[0..n-1])

//该算法用冒泡排序对给定的数组排序

//输入：一个可排序数组A[0..n-1]

//输出：升序排序的数组A[0..n-1]

for i←0 to n-2 do

for j←0 to n-2-i do

if A[j+1]< A[j]

swap A[j] and A[j+1]

**问题：**

①. 冒泡排序稳定吗？ 稳定

②. 请证明,如果对列表比较一遍之后没有交换元素的位置,那么这个表已经排好序了,算法可以停止了。

③. 结合所做的改进,为冒泡排序写一段伪代码。

④. 请证明改进的算法最差效率也是平方级的。

4. 以下是蛮力字符串匹配算法：

算法 BruteForceStringMatch(T[0..n-1],P[0..m-1])

//该算法实现了蛮力字符串匹配

//输入：一个n个字符的数组T[0..n-1],代表一段文本，一个m个字符的数组P[0..m-1],代表一个模式

//输出：如果查找成功，返回文本的第一个匹配子串中第一个字符的位置，否则返回-1.

for i←0 to n-m do

j←0

while j<m and P[j]=T[i+j] do

j←j+1

if j=m return i

return -1

**问题：**

①. 实现蛮力字符串匹配算法，输入实例：T=”abcaacbacacdfgcds”,P=”acb”,输出4。

②. 在算法中加入一个计数器，如果要在下面的文本中查找模式“GANDHI“。则蛮力算法将要执行的字符的比较次数是多少？

THERE IS MORE TO LIFE THEN INCREASEING ITS SPEED

假设查找前已知文本的长度，在这里它是47个字符。

③. 修改算法，求出给定文本中匹配的子串的个数。例如：输入实例：T=”acdaacdacacdfgcds”,P=”acb”,输出3。

④. 修改算法，使算法能够实现求在一段给定的文本中查找以A开始、以B结束的子串的数量(例如，在CABAAXBYA中，有4个这样的子串)。

思考：在第③中，你能实现把子串”acb”替换成“xyz“吗？

5. 请对凸包问题进行建模，并写一个程序，实现凸包问题的蛮力算法。

6. 应用穷举查找，把书中做了一个开头的分配问题的实例补充完整，并写出一个程序，实现分配问题的蛮力算法。请查阅资料了解匈牙利方法解决任务分配问题。

7. 设计一个程序，对于给定的图，它能够输出：

a. 图的顶点，按照被DFS遍历第一次访问到的先后次序；

b. 图的顶点，按照被BFS遍历访问到的先后次序。

1. **注意事项**
2. 注意要保证枚举了每一种情况，不会有遗漏的情形发生，这是枚举算法正确性的保证。
3. 注意在枚举时要保证每种情形只枚举一次，尽量避免重复的计算，这样才能保证时间复杂度在可以接受的范围内。