### 1. Fibonacci sequence通项公式

Fibonacci数列为1，1，2，3，5，8，13，……形式的数列，其每一项的数值等于前面两项的数值之和。设此数列中第n项为为，则有

其中n=1,2,3,…，，

构造成矩阵相乘形式为

当n=1时，上式可以写成

同理，当n=2,3,4,…时，可以写出类似的递推形式。

记矩阵A=，上述写出的等式组里，每个等式都可以由前一个等式左乘矩阵A得到，所以易归纳得到

其中，即

所以只要计算即可得到的表达式。更精细的说，由于不需要得到等式左边向量第一行的，等式最右边向量的第二行为0，故只要计算的第二行第一列即可。

为了便于归纳计算，将其进行特征值分解，A为实对称矩阵且其行列式不为0，所以矩阵A可以写成

其中U的列向量由A的特征向量构成，为相应特征值构成的对角矩阵，为U的逆矩阵。

将上式不断左乘A，易知

所以只要解得A的特征值与特征向量即可得到的第二行第一列，进而得到。

由写出

解得特征值，

代入，求基础解系得到相应的特征向量

对应的特征向量可以表示为，其中k≠0。

对应的特征向量可以表示为，其中k≠0。

由于k值在分解计算中与其倒数相乘抵消，不影响最终结果，故令k=1，写出一个U

其行列式为，由伴随矩阵除以行列式得到

代入，得

代回之前的递推式得

### 2.复杂度从小到大排序

O(1)

O(log N)

O(N)

O(Nlog N)

O(N^2)

O(N^{2.1})

O(N^2log N)

O(2^N)

O(3^N)

O(N!)

### 3. 字符串Levenshtein距离的动态规划计算 python

程序文件为20200519053-work1.py 运行提示输入两个字符串，输出Levenshtein距离

# -\*- coding: UTF-8 -\*-

def edit\_dist(stra, strb):

arra = []

arrb = []

for c in stra:

arra.append(c)

for c in strb:

arrb.append(c)

lena = len(arra)

lenb = len(arrb)

lenboud = max(lena,lenb)

arrf = [[0 for i in range(0,lenboud+2)]for i in range(0,lenboud+2)]

for i in range(1,lenboud+2):

arrf[i][0]=i

arrf[0][i]=i

for i in range(1,lena+1):

for j in range(1,lenb+1):

if (arra[i-1]==arrb[j-1]):

arrf[i][j]=arrf[i-1][j-1]

else:

arrf[i][j]=min(arrf[i][j-1],min(arrf[i-1][j],arrf[i-1][j-1]))+1

print('Levenshetein string distance = ',arrf[lena][lenb])

str1 = input('input first string:')

str2 = input('input second string:')

edit\_dist(str1,str2)

### 4. 非技术问题

Q1

没做过

Q2

我的公司栗子树科技目前为新零售和物联网方面的解决方案开发商，开发所应用的多为传统方法（之前没有自行开发AI领域的东西），之前有类似图像识别之类的需求，也都是购买现成的第三方库或者服务来实现，未来想做自己的产品，所以想先广泛了解各种技术的具体实现原理，以便于未来决策。之前看到课程大纲介绍感觉这个课程比市面上的其他课程更贴近本质方法而不是简单的复制一堆案例，所以选择了这个课程。

Q3

各种算法的本质原理和细节，剖析的越深入详细越好。

了解AI领域最新进展，最好能有近期相对重要的论文的解读和实验复现。