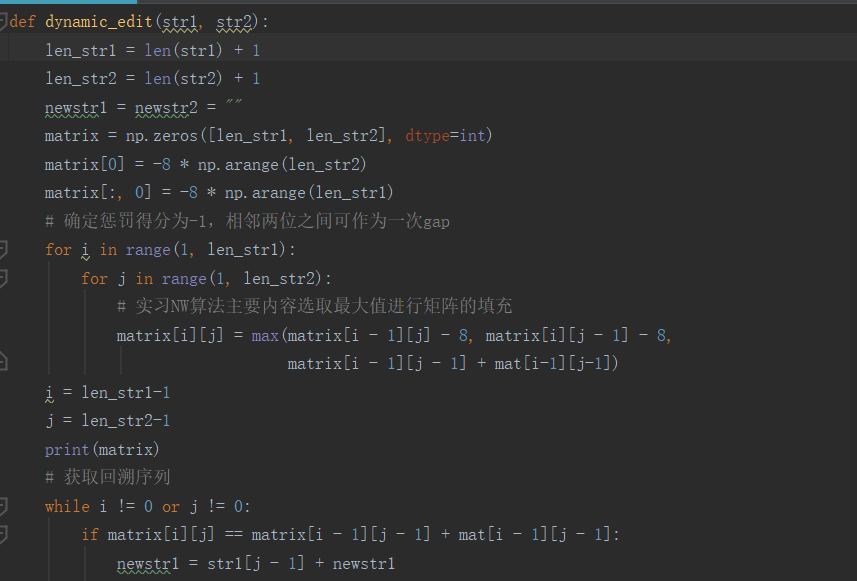
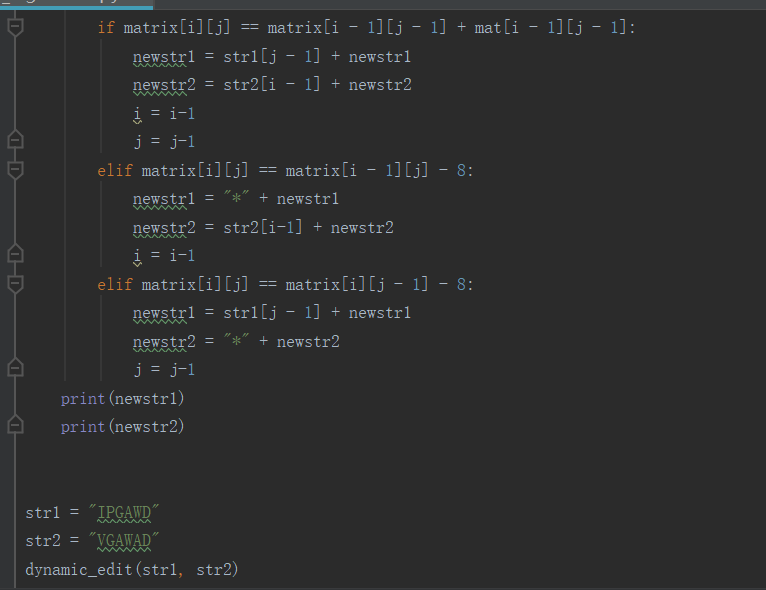
NW算法python实现

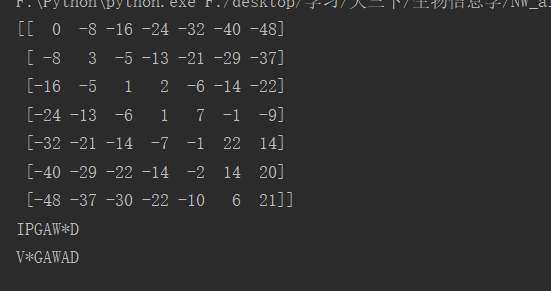
代码如下







import numpy as np  
  
  
# 导入Blosum矩阵进行罚分计算  
mat = np.zeros((6, 6), dtype=int)  
with open("Blosum\_m.txt", "r")as f:  
 lines = f.readlines()  
 for i in range(len(lines)):  
 lines[i] = lines[i].strip("\n").split(" ")  
 mat[i] = lines[i]  
  
  
def dynamic\_edit(str1, str2):  
 len\_str1 = len(str1) + 1  
 len\_str2 = len(str2) + 1  
 newstr1 = newstr2 = ""  
 matrix = np.zeros([len\_str1, len\_str2], dtype=int)  
 matrix[0] = -8 \* np.arange(len\_str2)  
 matrix[:, 0] = -8 \* np.arange(len\_str1)  
 # 确定惩罚得分为-1，相邻两位之间可作为一次gap  
 for i in range(1, len\_str1):  
 for j in range(1, len\_str2):  
 # 实习NW算法主要内容选取最大值进行矩阵的填充  
 matrix[i][j] = max(matrix[i - 1][j] - 8, matrix[i][j - 1] - 8,  
 matrix[i - 1][j - 1] + mat[i-1][j-1])  
 i = len\_str1-1  
 j = len\_str2-1  
 print(matrix)  
 # 获取回溯序列  
 while i != 0 or j != 0:  
 if matrix[i][j] == matrix[i - 1][j - 1] + mat[i - 1][j - 1]:  
 newstr1 = str1[j - 1] + newstr1  
 newstr2 = str2[i - 1] + newstr2  
 i = i-1  
 j = j-1  
 elif matrix[i][j] == matrix[i - 1][j] - 8:  
 newstr1 = "\*" + newstr1  
 newstr2 = str2[i-1] + newstr2  
 i = i-1  
 elif matrix[i][j] == matrix[i][j - 1] - 8:  
 newstr1 = str1[j - 1] + newstr1  
 newstr2 = "\*" + newstr2  
 j = j-1  
 print(newstr1)  
 print(newstr2)  
  
  
str1 = "IPGAWD"  
str2 = "VGAWAD"  
dynamic\_edit(str1, str2)



可得输出得分矩阵为

[[ 0 -8 -16 -24 -32 -40 -48]

[ -8 3 -5 -13 -21 -29 -37]

[-16 -5 1 2 -6 -14 -22]

[-24 -13 -6 1 7 -1 -9]

[-32 -21 -14 -7 -1 22 14]

[-40 -29 -22 -14 -2 14 20]

[-48 -37 -30 -22 -10 6 21]]

所得匹配结果为：

IPGAW\*D

V\*GAWAD