算法设计与分析实验报告

学号：12017001973 姓名：王亚歌

## 一、实验内容（问题）

★ 问题描述：对于长度相同的两个字符串A和B，其距离定义为相应位置字符距离之和。两个非空格字符的距离是它们的ASCII码之差的绝对值；空格与空格的距离为0，空格与其他字符的距离为一定值k。

在一般情况下，字符串A和B的长度不一定相同。字符串A的扩展是在A中插入若干空格字符所产生的字符串。在字符串A和B的所有长度相同的扩展中，有一对距离最小的扩展，该距离称为字符串A和B的扩展距离。

对于给定的字符串A和B，试设计一个算法，计算其扩展距离。

★ 算法设计：对于给定的字符串A和B，计算其扩展距离。

★ 数据输入：由文件input.txt给出输入数据。第1行是字符串，第2行是字符串B，第3行是空格与其他字符的距离定值k。

★ 结果输出：将计算出的字符串A和B的扩展距离输出到文件output.txt

输入文件示例 输出文件示例

input.txt output.txt

cmc 10

snmn

2

## 二、算法描述

算法思想描述：设字符串和的字串和B[1...j]的扩展距离是；

依题意，字符串和有三种可能的情况：

1）A串最后一个字符是空格，串最后一个字符是字母，则；

2）A串最后一个字符时字母，串最后一个字符时空格，则；

3）串和串最后一个字符均是字母，则；

由上可知，具有最优子结构性质，且满足如下递推式：

从动态规划递归式可知，算法的时间复杂度为，和分别是字符串和的长度。

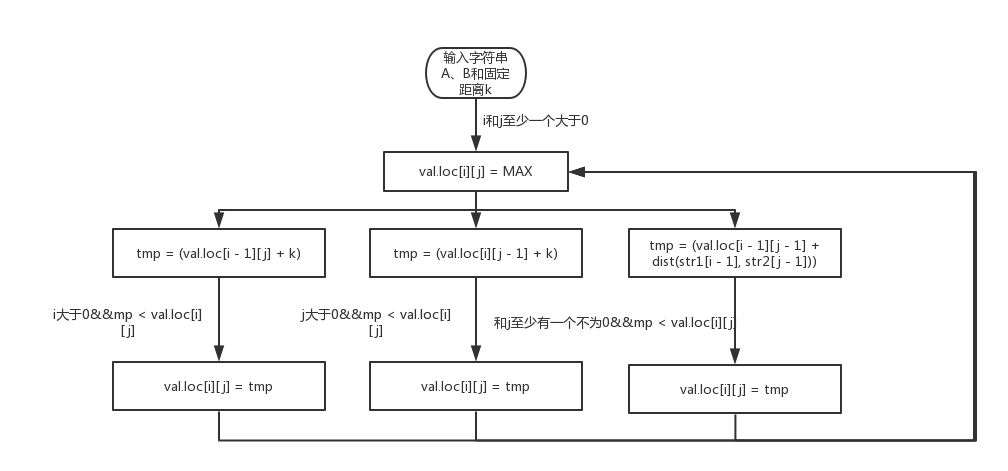
函数的作用：

dist(a,b)：返回字符a与b的ASCII码的差的绝对值

参数 a,b为输入的字符串A、B

comp()主要算法，使用动态规划算法，自底向上的计算各个子问题并利用每次计算的结果，避免重复运算

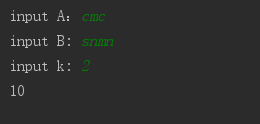
## 三、详细设计



## 四、程序代码

#!/usr/bin/env python  
# -\*- coding:utf-8 -\*-  
import numpy as np  
import pandas as pd  
  
MAX = 100000 #标识最大的可能整数  
def dist(a, b): #返回字符a与b的ASCII码的差的绝对值  
 return abs(ord(a) - ord(b))  
val = pd.DataFrame(range(0), index=range(-1, 300), columns=range(-1, 300))  
def comp():  
 val.loc[0][0] = 0;  
 len1 = len(str1)  
 len2 = len(str2)  
 for i in range(0, len1+1):  
 for j in range(0, len2+1):  
 if (i + j): #i和j至少一个大于0  
 val.loc[i][j] = MAX  
  
 tmp = (val.loc[i - 1][j] + k)  
 if (i)and(tmp < val.loc[i][j]): #i大于0  
 val.loc[i][j] = tmp  
  
 tmp = (val.loc[i][j - 1] + k)  
 if (j)and(tmp < val.loc[i][j]): #j大于0  
 val.loc[i][j] = tmp  
  
 tmp = (val.loc[i - 1][j - 1] + dist(str1[i - 1], str2[j - 1]))  
 if (i \* j)and (tmp < val.loc[i][j]): #i和j至少有一个不为0  
 val.loc[i][j] = tmp  
  
 print val.loc[len1][len2]  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 str1 = raw\_input('input A：') #字符串A  
 str2 = raw\_input('input B: ') #字符串B  
 k = input('input k: ') #与空格固定距离k  
 comp()

## 五、测试和结果



输入字符串A:cmc

输入字符串B：snmn

输入固定距离k:2

输出扩展距离：10

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | c | 空格 | 空格 | m | 空格 | c |
| B | 空格 | s | n | m | n | 空格 |
| 字符距离 | 2 | 2 | 2 | 0 | 2 | 2 |