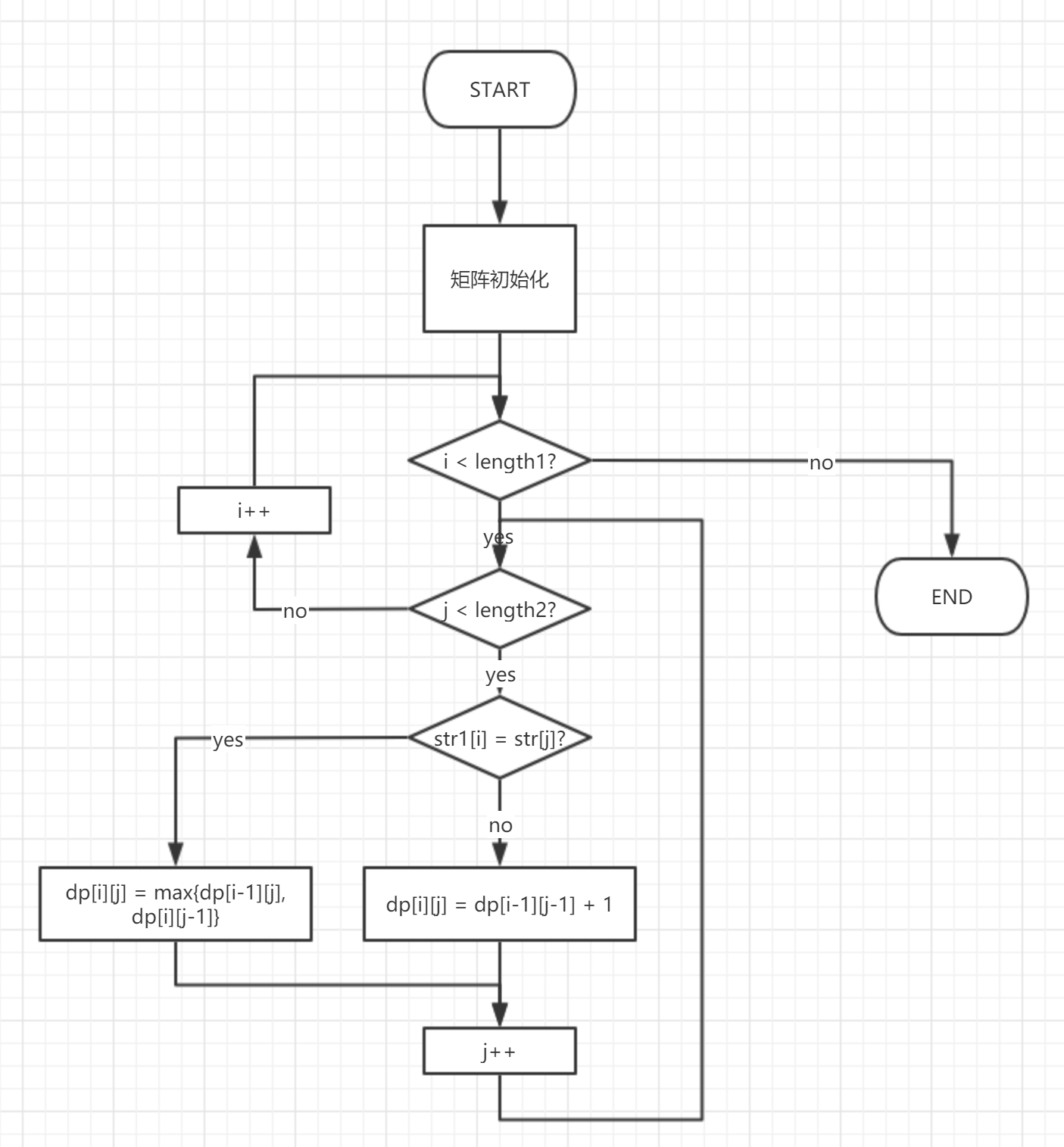
**软件安全实验三 恶意代码特征提取**

1170300728-汤添凝

1. **设计的算法流程图，以及算法说明**



**算法说明：**

**①.DP矩阵初始化：**

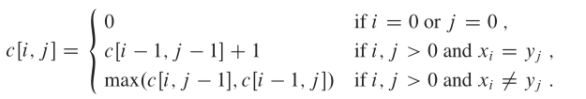
初始化dp矩阵

如果s1[0]!=s2[j] ,dp[0][j] = 0，否则dp[0][j]=1

如果s1[i]!=s2[0] ,dp[i][0] = 0，否则dp[i][0]=1

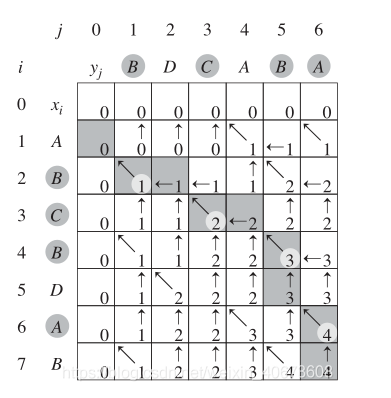
**②.计算DP矩阵**

依据如下递归方程：计算dp矩阵



**③.路径回溯（输出）**

最后回溯dp矩阵，深度优先遍历，输出所有符合条件的LCS，具体路径回溯如下图：



1. **关键的数据结构，及简单说明**

关键数据结构：

String str1,str2,

功能：表示两个字符串

int dp[][]

功能：表示dp矩阵

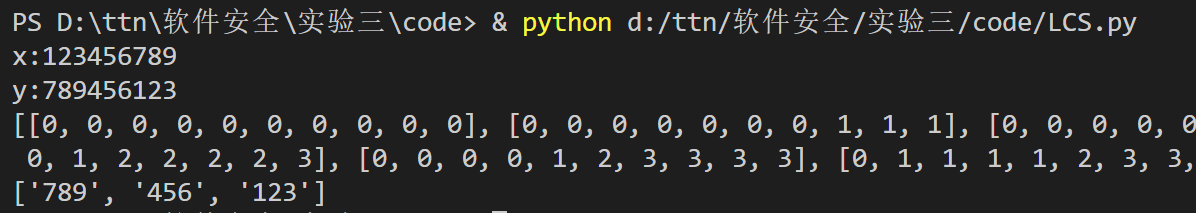
int length1

功能：表示str1的长度

int length2

功能：表示str2的长度

1. **实验结果的截图**



1. **实验代码：**

代码语言：Python

def merge(x, y, lcslist, i, j, output, outputlist):

    temp = lcslist[i][j]

    flag = False

    if temp == 0:

        output = output[::-1]

        if output not in outputlist:

            outputlist.append(output)

        return

    if lcslist[i][j-1] == temp:

        merge(x,y,lcslist,i,j-1,output,outputlist)

        flag = True

    if lcslist[i-1][j] == temp:

        merge(x,y,lcslist,i-1,j,output,outputlist)

        flag = True

    if flag == False:

        output += x[i-1]

        merge(x,y,lcslist,i-1,j-1,output,outputlist)

x = input("x:")

y = input("y:")

m = len(x)

n = len(y)

lcslist = [[0 for i in range(n + 1)] for j in range(m + 1)]

for i in range(m):

    for j in range(n):

        if x[i] == y[j]:

            lcslist[i+1][j+1] = lcslist[i][j] + 1

        elif lcslist[i+1][j] >= lcslist[i][j+1]:

            lcslist[i+1][j+1] = lcslist[i+1][j]

        else:

            lcslist[i+1][j+1] = lcslist[i][j+1]

print(lcslist)

output=""

outputlist=[]

merge(x, y, lcslist, m, n, output, outputlist)

print(outputlist)