```
"""
        Name: Ruiqing Qiu
        PID: A98022702
        Userid: rqiu
        filename: hw5.py
"""
import math
import matplotlib.pyplot as plt
aList = []
bList = []
observation = []
PI = []
LStarMatrix = []
with open ('transitionMatrix.txt') as transFile:
  for line in transFile:
    tmp = line.split()
    aList.append(tmp)
#print aList

with open ('emissionMatrix.txt') as emissionFile:
  for line in emissionFile:
    tmp = line.split()
    bList.append(tmp)
#print bList

with open ('observations.txt') as observationFile:
  for line in observationFile:
    tmp = line.split()
    observation = tmp
#print observation
with open ('initialStateDistribution.txt') as initial:
  for line in initial:
    tmp = line.split()
    PI.append(tmp[0])
#print PI

def calculateFirstColumn():
  ColList = []
  for i in range(0, 26):
    Oindex = int(observation[0])
    value = math.log(float(PI[i])) + math.log(float(bList[i][Oindex]))
    ColList.append(value)
  return ColList
def maxVal(targetList):
  maxVal = targetList[0]
  index = 0
  for i in range(0, 26):
    if maxVal < targetList[i]:
      maxVal = targetList[i]
      index = i
  return index

def calculateNextColumn(t, previous):
  ColList = []
  Oindex = int(observation[t])
  #loop through the current column
  for curr in range(0, 26):
    # using all previous 26 values to get a max value
    tmpList = []
    for prev in range(0, 26):
      #value = float(previous[prev])+math.log(float(aList[curr][prev])) + math.log(f
loat(bList[curr][Oindex]))
      value = float(previous[prev]) + math.log(float(aList[curr][prev]))
      tmpList.append(value)
    index = maxVal(tmpList)
    ColList.append(tmpList[index] + math.log(float(bList[curr][Oindex])))
  return ColList

def backtrackMax(targetList, t, j):
  #For the last T
  if t == 67999:
    maxValue = targetList[0]
    for i in range(1, 26):
```

```
      value = targetList[i]
      if maxValue < value:
        maxValue = value
        index = i
    return index
  else:
    maxValue = targetList[0] + math.log(float(aList[0][j]))
    index = 0
    for i in range (1, 26):
      value = targetList[i] + math.log(float(aList[i][j]))
      if maxValue < value:
        maxValue = value
        index = i
    return index
def backtrack():
  letterList = []
  previousIndex = backtrackMax(LStarMatrix[67999], 67999, 0)
  letterList.append(previousIndex)
  for t in range (67998, 0,-1):
    #value = maxSequence[t][maxVal(LStarMatrix[t])]
    previousIndex = backtrackMax(LStarMatrix[t], t, previousIndex)
    letterList.append(previousIndex)
    #letterList.append(maxSequence[t][backtrackMax(LStarMatrix[t],t)])
  letterList.reverse()
  return letterList

# The base case for the first column
listtmp = calculateFirstColumn()
print listtmp
LStarMatrix.append(listtmp)
#newList = calculateNextColumn(1, listtmp)
#print newList
for t in range (1, 68000):
  listtmp = calculateNextColumn(t, listtmp)
  LStarMatrix.append(listtmp)
  print t
finalList = backtrack()
print finalList

plt.plot(finalList)
plt.ylabel('letter')
plt.show()
```