Quiz4 Report

```
train=train.replace(" ","")
dict_3={}
for i in range(len(train)-2):
    substr=train[i:i+3]
    if substr not in dict_3:
        dict_3[substr]=train.count(substr)
dict_2={}
for i in range(len(train)-1):
    substr=train[i:i+2]
    if substr not in dict_2:
        dict_2[substr]=train.count(substr)
```

First, we construct the training data set using the given training data. For bigrams and trigrams, I construct two dictionary to record each pair's frequency which had appear in the training data.

```
if(Diff1<Diff2):
    element=7
else:
    element=11

Cipher=['']*element
j=0
for i in cipher:
    Cipher[int(j/11)]+=i
    j+=1
```

Use Quiz2 method to determine the ciphertext rectangles than distribute the char into correct column

```
for i in range(5):
    max=0 #用來記錄最大的conditional probability
    next_index=0 #用來記錄最後要排進ans的下個column對應的column index
    for column in range(7):
        num=0 #用來記錄每個column的conditional probability
        plain=Cipher[ans[i]][0]+Cipher[ans[i+1]][0] #bigram word
        if plain in dict_2:
            base=dict_2[plain] #bigram frequency
        else:
            base=0
        for index in range(11):
            if base!=0: #確保分母不為0
                if used[column]==False: #確定column沒有被排過
                    plain+=Cipher[column][index] #trigram word
                    if plain in dict_3:
                        pf=dict_3[plain] #trigram frequency
                        num+=(pf/base) #conditional probability
                    if index!=10:
                        plain=Cipher[ans[i]][index+1]+Cipher[ans[i+1]][index+1] #換下一row的前兩個字
                        if plain in dict_2:
                            base=dict_2[plain]
                        else:
                            base=0
        if max<num:
            next_index=column
            max=num
    ans[i+2]=next_index #記錄下個ans
    used[next_index]=True #確保不會再排到此column
```

Use for loop to calculate the conditional probability, the three layers respectively represent: first layer, there are totally 7 column with 11 element, for each three element construct a pair, we only need to do five rounds; second layer, every time do seven rounds to travel the column that choose the maximum conditional probability, and the used[] array would record which column had already been used; third layer, travel 11 row in each column to calculate trigram frequency, bigram frequency and conditional probability.

```
127  if element==7:
128      index=11 #反轉後的index
129  else:
130      index=7
131  ciph=['']*index
132  for i in range(index):
133      for order in ans:
134          ciph[i]+=Cipher[order][i] #將Cipher的column轉成row的形式
135  for i in range(index):
136      for word in ciph[i]:
137          print(word,end='')
138      print('\n')
```

Change the column into row then print answer..

```
GREECEA

NNOUNCE

DYESTER

DAYITHA

DREACHE

DAGREEM

ENTWITH

TURKEYT

OENDTHE

CYPRUSC

RISISNS
```