250 HW7 Q1

November 17, 2018

In [2]:
```
import numpy as np
import math
import matplotlib.pyplot as plt
import pandas as pd
```

In [3]:
```
# loading the parameter matrices
bik = pd.read_csv('emissionMatrix.txt', sep="\t", header=None).values
aij = pd.read_csv('transitionMatrix.txt', sep=" ", header=None)
aij = aij.drop([27], axis=1).values
pi_i = pd.read_csv('initialStateDistribution.txt', header=None).values
observation = pd.read_csv('observations.txt', sep=" ", header=None)
observation = observation.drop([325000], axis=1).values
```

In [42]:
```
def computeL(aij,bik,pi_i,observation):
    T = observation.shape[1]
    n = aij.shape[1]
    Lit = np.zeros((n,T))
    Lit[:,0]=np.squeeze(np.log(pi_i))+np.log(bik[:,observation[0,0]])
    for t in np.arange(0,T-1):
        for j in np.arange(n):
            tmp = Lit[:,t]+np.log(aij[:,j])
            Lit[j,t+1]=np.max(tmp)+np.log(bik[j,observation[0,t+1]])
    return Lit
```

In [43]:
```
Lit = computeL(aij,bik,pi_i,observation)
```

In [48]:
```
def ComputeViterbi(lit,aij):
    T = lit.shape[1]
    n = lit.shape[0]
    s = np.zeros((T,1), dtype=int)
    #base case
    s[T-1,0] = np.argmax(lit[:,T-1])
    # recursion case
    for i in np.arange(T-2,-1,-1):
        s[i,0] = phi(s[i+1,0],lit[:,i],aij)
    return s
```

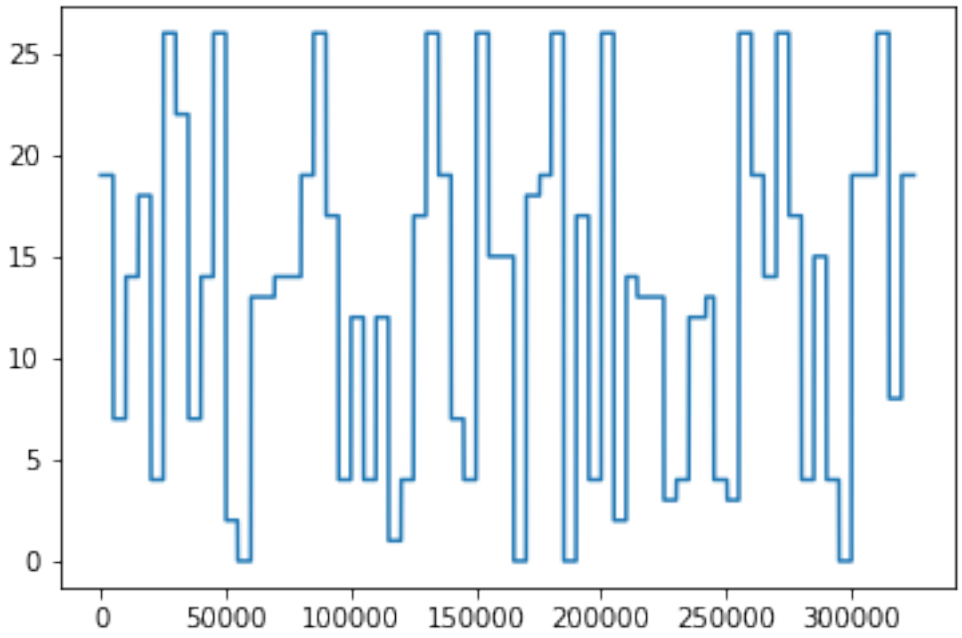
In [49]:
```
def phi(s,lit,aij):
    n = aij.shape[0]
```

```
            temp = np.ones((n,1))
            for i in range(n):
                temp[i,0] = lit[i] + np.log(aij[i,s])
            return np.argmax(temp)
```

```
In [50]: s = ComputeViterbi(Lit,aij)
```

```
In [52]: plt.plot(s)
```

```
Out[52]: [<matplotlib.lines.Line2D at 0x108bf9f28>]
```




```
In [94]: s = np.squeeze(s)
         alphabet=['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S'
         s_idx = s[np.diff(np.concatenate([[0],s]))!=0]
         decode = [alphabet[j] for j in s_idx]
```

```
In [106]: print(decode)
```

```
['T', 'H', 'O', 'S', 'E', ' ', 'W', 'H', 'O', ' ', 'C', 'A', 'N', 'O', 'T', ' ', 'R', 'E', 'M'
```