BCH 571 Bioinformatics for Life Scientists

Lab 10

For this lab, you will implement a dynamic programming algorithm to estimate the state of the coin (Fair or Biased) for the fair bet casino. This will require two arrays (lists), VF_i and VB_i, where VF_i is the maximum probability of the path up to flip i (1-indexed) ends in state F. Likewise, VB_i is the maximum probability of the path up to flip i (1-indexed) ends in state B The main recursion is:

$$\mathrm{VF_i} = \mathrm{P_{x_i}} \max \begin{cases} 0.9\ \mathrm{VF_{i-1}} \\ 0.1\ \mathrm{VB_{i-1}} \end{cases}$$

$$\mathrm{VB_i} = \mathrm{P_{x_i}} \max \begin{cases} 0.9\ \mathrm{VB_{i-1}} \\ 0.1\ \mathrm{VF_{i-1}} \end{cases}$$

where $\mathrm{P_{x_i}}$ is the probability of getting coin flip result x_i. Initialize $VF_0 = VB_0 = 0.5$.

Here is the pseudocode (where strings are 1-indexed):

Provide string x, the sequence of H and T for heads and tails and output string pi, the sequence of coins being used for the maximum probability path.
Provide also TP, the probability of transitioning back to the same state (0.9 by default)

```
FairCasino(x,TP)
  Make arrays VF and VB for length(x)+1 states, 0-indexed
  VF_0 ← 0.5
  VB_0 ← 0.5
  for i=0 to length(x)
   if (xi = H)
      Px ← 0.75
   else
       Px ← 0.25
  VF_i ← 0.5 × max(TP × VF_i-1, (1-TP) × VB_i-1)
  VB_i ← Px × max(TP × VB_i-1, (1-TP) × VF_i-1)

  Make empty string pi
  i ← length(x)
  if (VF_i > VB_i)
    state ← "F"
    pi ← "F"
  else
    state ← "B"
    pi ← "B"
  while(i>1)
     if (state = "F")
      if (VF_i = 0.5 × (TP × VF_i-1))
         state ← "F"
         pi ← "F" + pi
      else
         state ← "B"
         pi ← "B" + pi
    else
          if (xi = "H")
             Px ← 0.75
          else
             Px ← 0.25
         if (VB_i = Px × (TP × VB_i-1))
            state ← "B"
            pi ← "B" + pi
         else
            state ← "F"
            pi ← "F" + pi
    i ← i -1
  return pi
```

Note that the parenthesis guarantee that the multiplication will be in the same order as the fill step, otherwise rounding might occur differently from fill and traceback.

Run the algorithm with the sequence:
x= HHHHTTTTHHHHHHHHTHTHTHTHHHHTHHTTTTHHTHHT