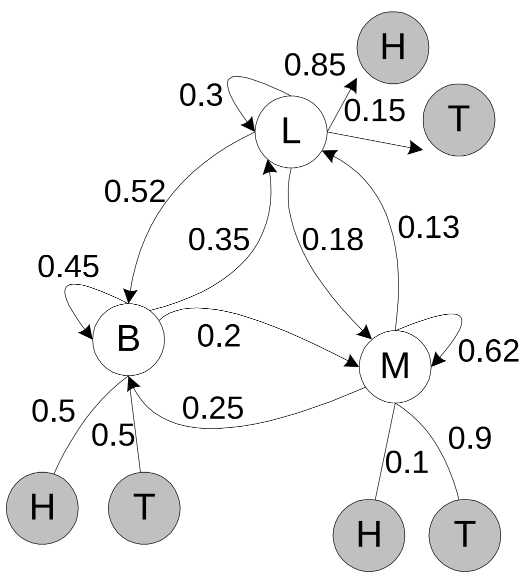
CSCI-581 Homework 5 Programming Viterbi's Algorithm

We would like to avoid the problem of *underflow*, so we are going to do all calculations in *log* space. Since *logs* of probabilities are negative numbers, and we wish to use positive numbers for convenience, we will modify the Viterbi's algorithm slightly: instead of finding the maximum of $\mathrm{P}(\mathrm{e}_{1:i}, \mathrm{X}_0, \mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_i)$, we'll find the minimum of $-log_2 P(e_{1:i}, X_0, X_1, X_2, \ldots, X_i)$ (negative log of P).

Experiment. Given the starting transition and sensory probabilities for an example of flipping a coin when three kinds of coins are used, namely, balanced B, loaded with higher probabilities of Heads L and loaded with higher probabilities of Tails M; find the most likely sequence of states that had generated the given sequence of observations.

Assume that $P(X_0 = x) = 1/3$ for $x = B, L, M$. $-\log_2(P(X_0 = x)) = 1.58496$.
Given sequence of observations: **HHHTHTTTTH**



Negative logs of transition probabilities:

	X_{i-1}=B	X_{i-1}=L	X_{i-1}=M
-log(P(B_i|X_{i-1}))	1.15200	0.94342	2.00000
-log(P(L_i|X_{i-1}))	1.51457	1.73697	2.94342
-log(P(M_i|X_{i-1}))	2.32193	2.47393	0.68966

Negative logs of sensory probabilities:

	-log(P(H_i|X_i))	-log(P(T_i|X_i))
X_i =B	1	1
X_i =L	0.23447	2.73697
X_i =M	3.32193	0.15200

Assignment:

(1) Implement the Viterbi's algorithm for this experiment. Write functions that will fill in Probabilities table and Backtracking table. Write a recursive function that would collect the most likely sequence of states using Backtracking table.

(2) Given the actual sequence of states and the sequence of states calculated with the Viterbi's algorithm, calculate the Accuracy of the algorithm. Accuracy = correctly predicted states / length of the state sequence. The length of state sequence is one longer than the length of the given sequence of observations.

Input:

Line 1: integer N and M and L; N is total values for states, M is total values for observations, L is the length of sequence of observations.
Line 2: Total of N initial probabilities P(X0) (double).
Next N lines: N x N matrix T of transition probabilities (double). B is 0, L is 1, M is 2. Note that T[i][j] is P(State i at time t | State j at time t-1), i.e. the sum of probabilities in a column is equal to 1.
Next N lines: N x M matrix O of observation probabilities (double). Each row corresponds to a state. O[i][j] is P(observation j | state i).
Next line: L values of observations (in rage 0, 1, … , M-1) (In our example, Head is 0, Tail is 1).
Next line: (L + 1) values of actual states that generated the given sequence of observations (in range 0, 1, …, N-1).

Output:
Line 1: N double values, the last column's entries of Probability table; one value for each state, **with precision of 12 decimal digits after the decimal point,** in format:
<double><space>…<double><space><endl>
Line 2: total of (L + 1) states, the sequence of most likely states that generated the given sequence of observations, in format:
<state><space>…<state><space><endl>
Line 3: Accuracy (expressed using percentage with two decimal points) in format:
<Accuracy><endl>

Example:
Input Sample:
3 2 4
0.3333334 0.3333333 0.3333333
0.45 0.52 0.25
0.35 0.3 0.13
0.2 0.18 0.62
0.5 0.5
0.85 0.15
0.1 0.9
0 0 1 1
0 1 0 1 0

Probability table:
```
1.584962212182 3.528379116624 5.277417110283 7.220834014725 9.372837108170
1.584962644991 3.334000638649 5.277417543091 9.528955877279 11.472372781721
1.584962644991 5.596550619266 9.129859921869 7.751348298615 8.593011271448
```

Backtracking table:
```
0 1 1 1 0
1 0 0 0 0
2 2 1 0 2
```

Output Sample:
```
9.372837108170 11.472372781721 8.593011271448
0 1 0 2 2
60.00
```

Submission: submit *main.cpp* to HW5 on turnin.