# Checking a Casino

Fair coin:
Pr(Heads) = 0.5

Biased coin:
Pr(Heads) = 0.75

?

Suppose either a fair or biased coin was used to generate a sequence of heads & tails. But we don't know which type of coin was actual used.

Heads/Tails: ↑ ↑ ↓ ↓ ↓ ↓ ↑ ↑ ↑ ↑ ↓ ↑ ↓ ↑ ↓ ↑ ↓ ↑

# Checking a Casino



Fair coin:
Pr(Heads) = 0.5



Biased coin:
Pr(Heads) = 0.75

?

Suppose either a fair or biased coin was used to generate a sequence of heads & tails. But we don't know which type of coin was actual used.

Heads/Tails: ↑ ↑ ↓ ↓ ↓ ↓ ↑ ↑ ↑ ↑ ↓ ↑ ↓ ↑ ↓ ↑ ↓ ↑

# Checking a Casino



Fair coin:
Pr(Heads) = 0.5



Biased coin:
Pr(Heads) = 0.75

?

Suppose either a fair or biased coin was used to generate a sequence of heads & tails. But we don't know which type of coin was actual used.

Heads/Tails: ↑ ↑ ↓ ↓ ↓ ↓ ↑ ↑ ↑ ↑ ↓ ↑ ↓ ↑ ↓ ↑ ↓ ↑

How could we guess which coin was more likely?

# Compute the Probability of the Observed Sequence

Fair coin: Pr(Heads) = 0.5
Biased coin: Pr(Heads) = 0.75

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| X = | ↑ | ↑ | ↓ | ↓ | ↓ | ↓ | ↑ |
| Pr(x \| Fair) = | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| Pr(x \| Biased) = | 0.75 | 0.75 | 0.25 | 0.25 | 0.25 | 0.25 | 0.75 |

# Compute the Probability of the Observed Sequence

Fair coin: Pr(Heads) = 0.5
Biased coin: Pr(Heads) = 0.75

X = ↑ ↑ ↓ ↓ ↓ ↓ ↑

$\Pr(x \mid \text{Fair}) = 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 \quad = 0.5^7 = 0.0078125$

$\Pr(x \mid \text{Biased}) = 0.75 \times 0.75 \times 0.25 \times 0.25 \times 0.25 \times 0.25 \times 0.75 \quad = 0.001647949$

# Compute the Probability of the Observed Sequence

| |
|---|
| Fair coin: Pr(Heads) = 0.5 |
| Biased coin: Pr(Heads) = 0.75 |

$X = \uparrow \quad \uparrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \uparrow$

$\Pr(x \mid \text{Fair}) = 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 \quad = 0.5^7 = 0.0078125$

$\Pr(x \mid \text{Biased}) = 0.75 \times 0.75 \times 0.25 \times 0.25 \times 0.25 \times 0.25 \times 0.75 \quad = 0.001647949$

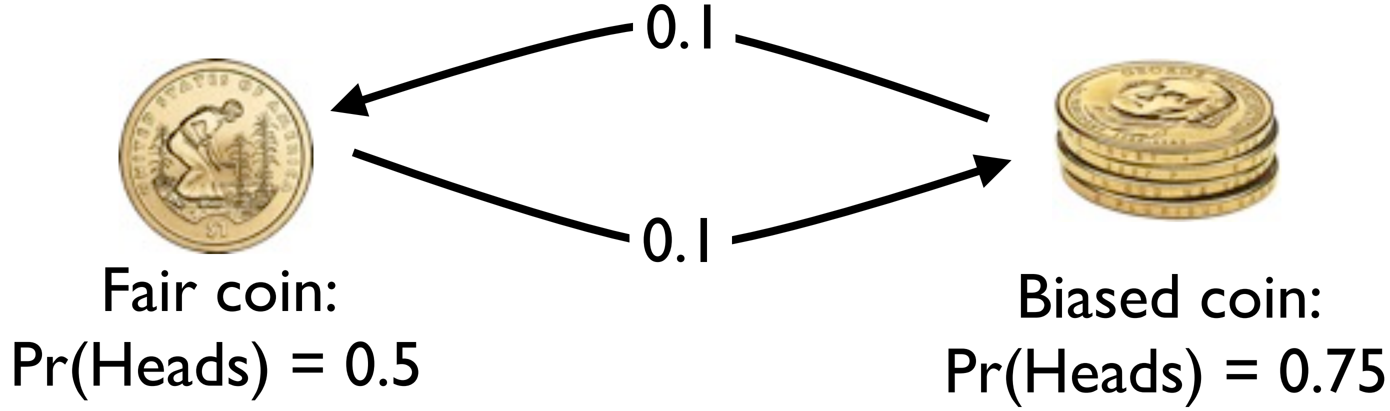
The *log-odds score:*

$$\log_2 \frac{\Pr(x \mid \text{Fair})}{\Pr(x \mid \text{Biased})} = \log_2 \frac{0.0078}{0.0016} = 2.245$$

> 0. Hence "Fair" is a better guess.

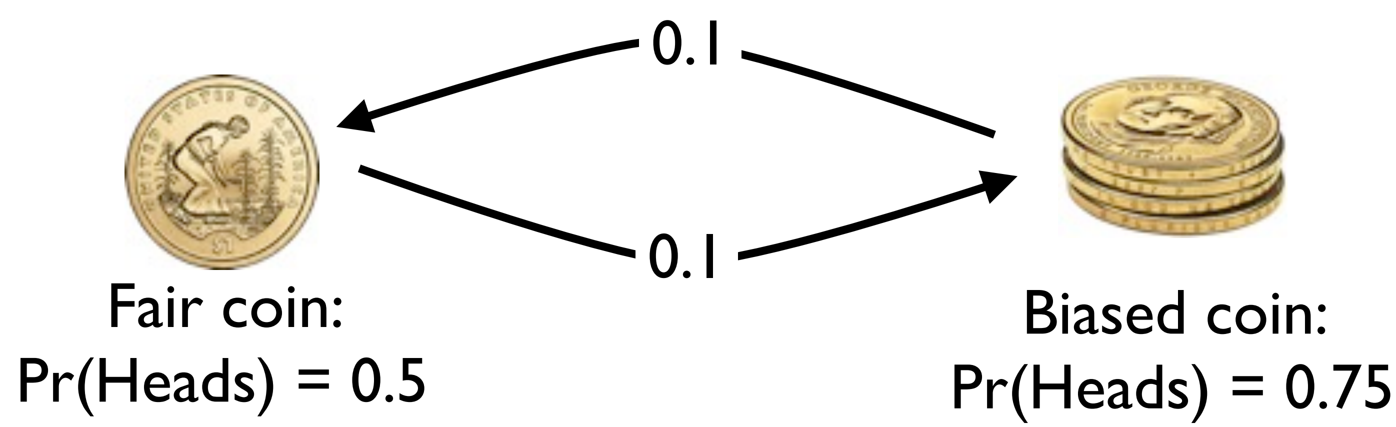
# What if the casino switches coins?

Fair coin:  Pr(Heads) = 0.5
Biased coin: Pr(Heads) = 0.75
Probability of switching coins = 0.1



# What if the casino switches coins?

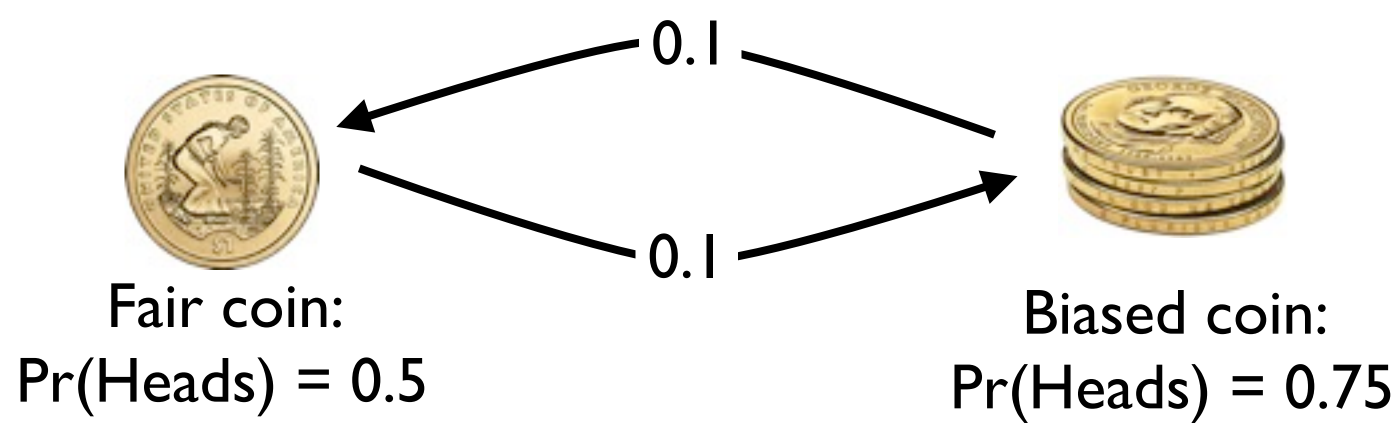
Fair coin:  Pr(Heads) = 0.5
Biased coin: Pr(Heads) = 0.75
Probability of switching coins = 0.1




How can we compute the probability of the entire sequence?

# What if the casino switches coins?

Fair coin:  Pr(Heads) = 0.5
Biased coin: Pr(Heads) = 0.75
Probability of switching coins = 0.1

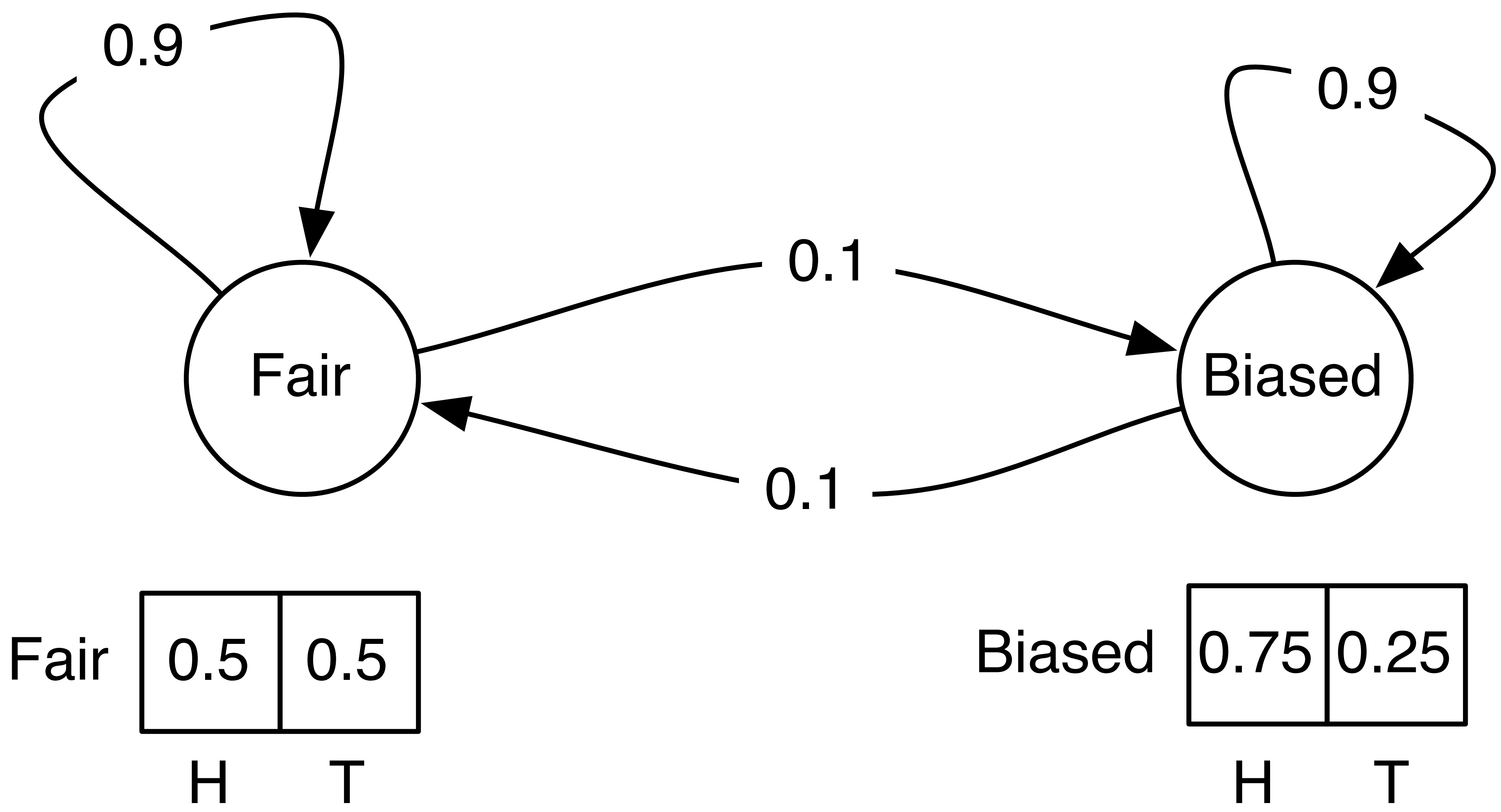



How can we compute the probability of the entire sequence?

How could we guess which coin was more likely **at each position**?

# Hidden Markov Model (HMM)

Fair coin:  Pr(Heads) = 0.5
Biased coin: Pr(Heads) = 0.75
Probability of switching coins = 0.1



# Formal Definition of a HMM

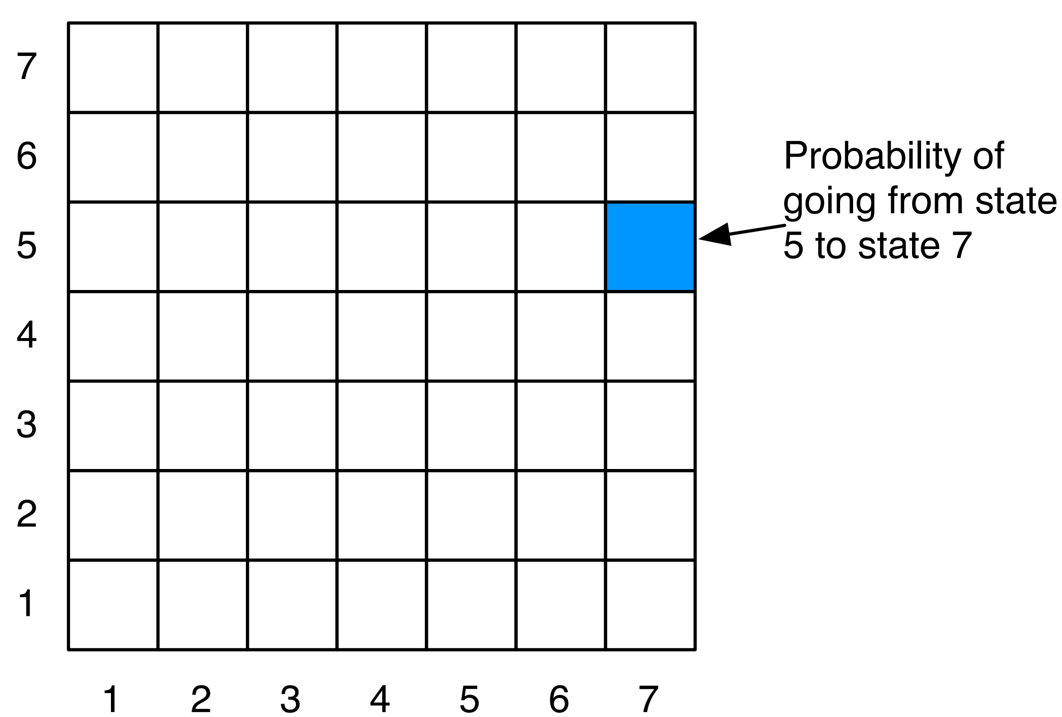
$\Sigma$ = alphabet of symbols.

Q = set of states.

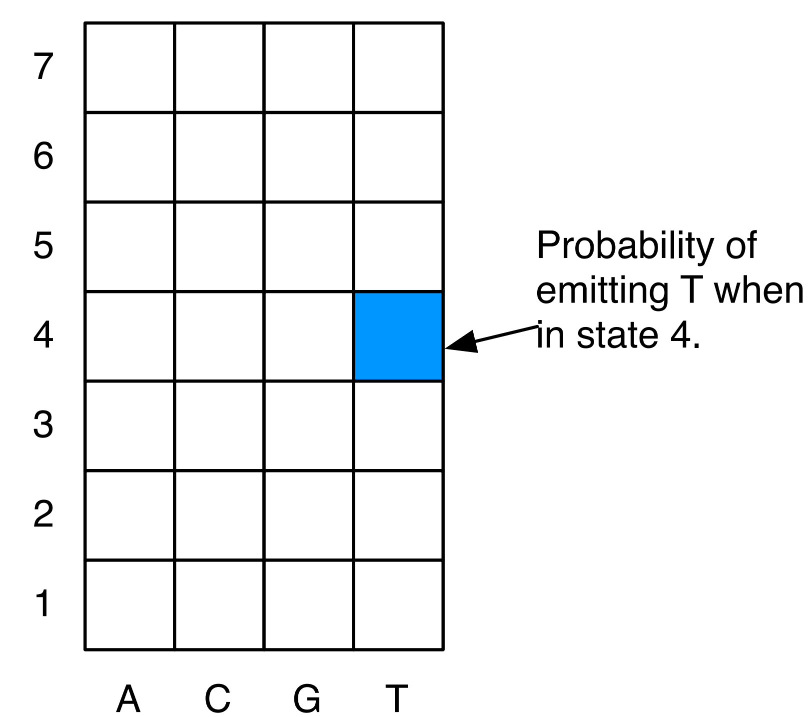
A = an $|Q| \times |Q|$ matrix where entry (k,l) is the probability of moving from state k to state l.

E = a $|Q| \times |\Sigma|$ matrix, where entry (k,b) is the probability of emitting b when in state k.

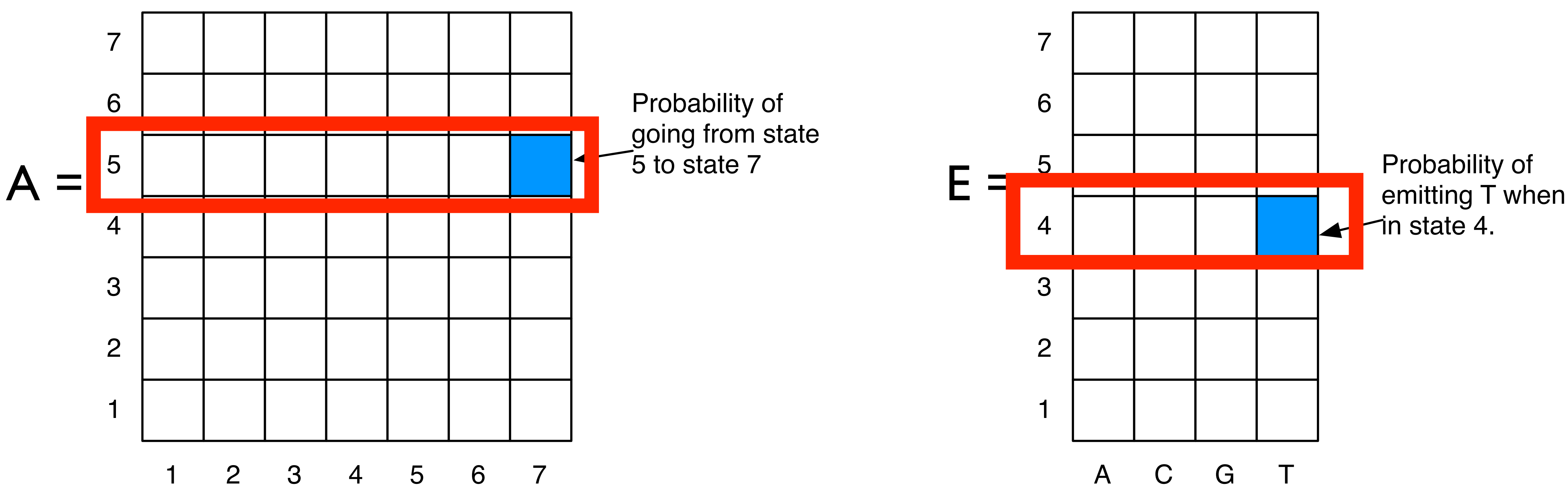
A =




E =

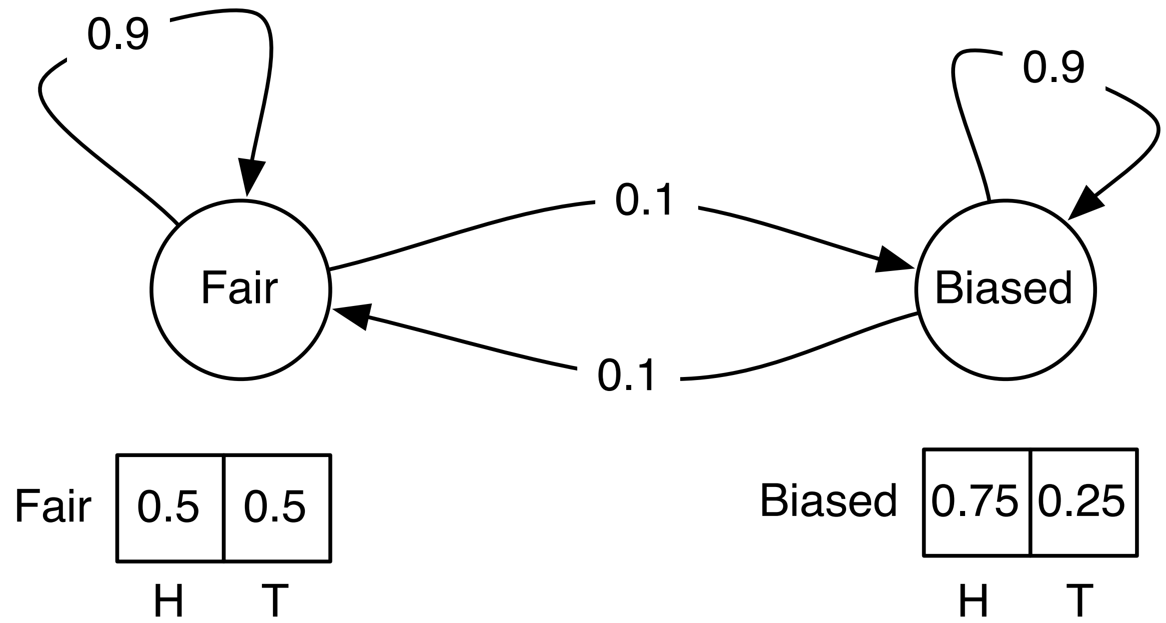


# Constraints on A and E




Sum of the # in each row must be 1.

# Computing Probabilities Given Path




| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $x =$ | ↓ | ↑ | ↓ | ↑ | ↑ | ↑ | ↓ | ↑ | ↑ | ↓ |
| $\pi =$ | F | F | F | B | B | B | B | F | F | F |
| $\Pr(x_i \mid \pi_i) =$ | 0.5 | 0.5 | 0.5 | 0.75 | 0.75 | 0.75 | 0.25 | 0.5 | 0.5 | 0.5 |
| $\Pr(\pi_i \rightarrow \pi_{i+1}) =$ | 0.1 | 0.9 | 0.9 | 0.1 | 0.9 | 0.9 | 0.9 | 0.1 | 0.1 | 0.1 |

# The Decoding Problem

Given x and π, we can compute:

- Pr(x | π): product of $\Pr(x_i \mid \pi_i)$
- Pr(π): product of $\Pr(\pi_i \to \pi_{i+1})$
- Pr(x, π): product of all the $\Pr(x_i \mid \pi_i)$ and $\Pr(\pi_i \to \pi_{i+1})$

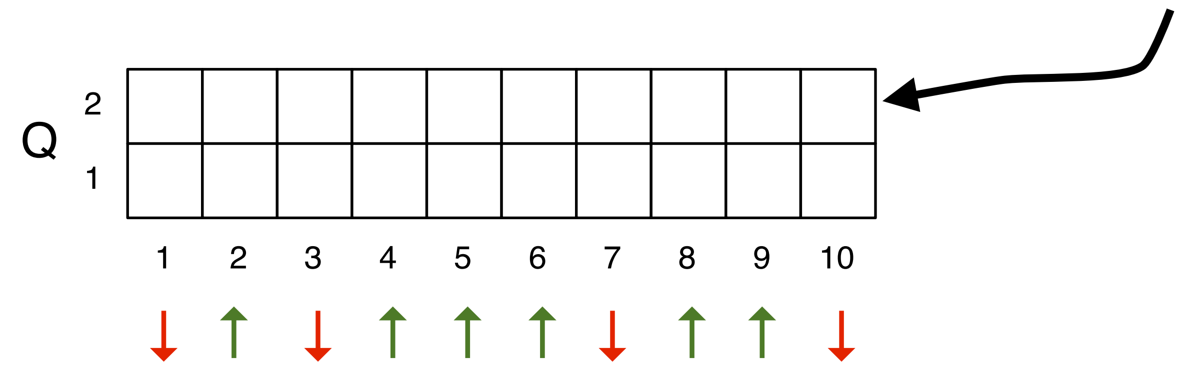
$$\Pr(x, \pi) = \Pr(\pi_0 \to \pi_1) \prod_{i=1}^{n} \Pr(x_i \mid \pi_i) \Pr(\pi_i \to \pi_{i+1})$$

But they are "hidden" Markov models because π is unknown.

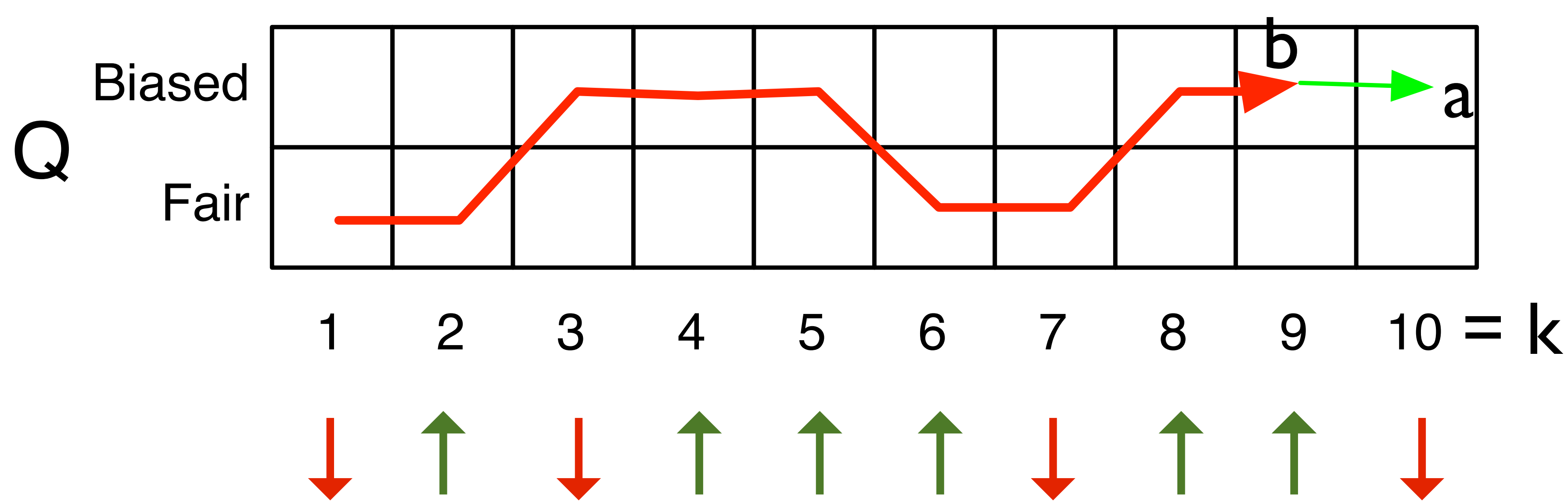
**Decoding Problem:** Given a sequence $x_1x_2x_3...x_n$ generated by an HMM (∑, Q, A, E), find a path π that maximizes Pr(x, π).

# The Viterbi Algorithm to Find Best Path

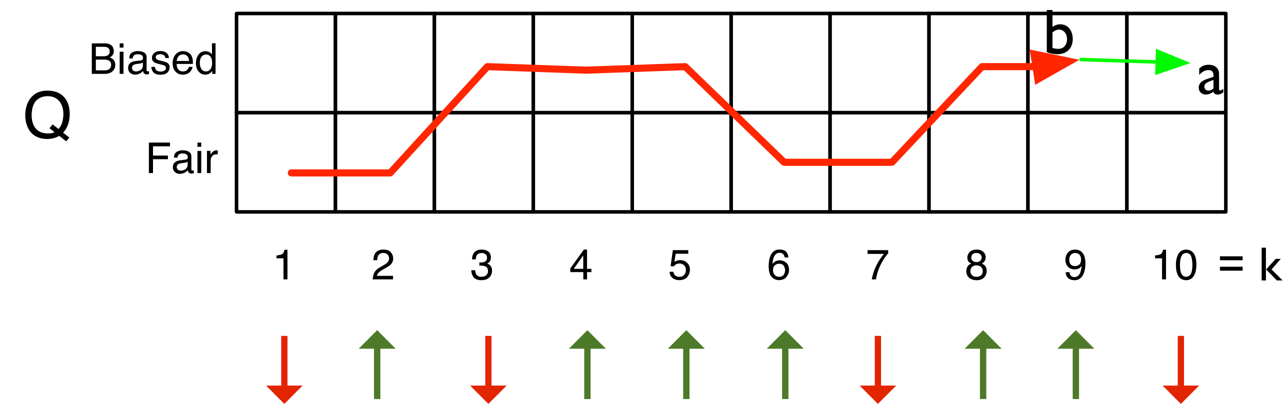
A[a, k] := the probability of the **best** path for $x_1 \ldots x_k$ that ends at state a.




A[a, k] = the path for $x_1 \ldots x_{k-1}$ that goes to some state *b* times cost of a transition from b to i, and then to output $x_k$ from state a.



# Viterbi DP Recurrence




$$A[a,k] = \max_{b \in Q} \{ \underbrace{A[b, k-1]} \times \underbrace{\Pr(b \to a)} \times \underbrace{\Pr(x_k \mid \pi_k = a)} \}$$

Over all possible previous states.

Best path for $x_1..x_k$ ending in state *b*

Probability of transitioning from state *b* to state *a*

Probability of outputting $x_k$ given that the *k*th state is *a*.

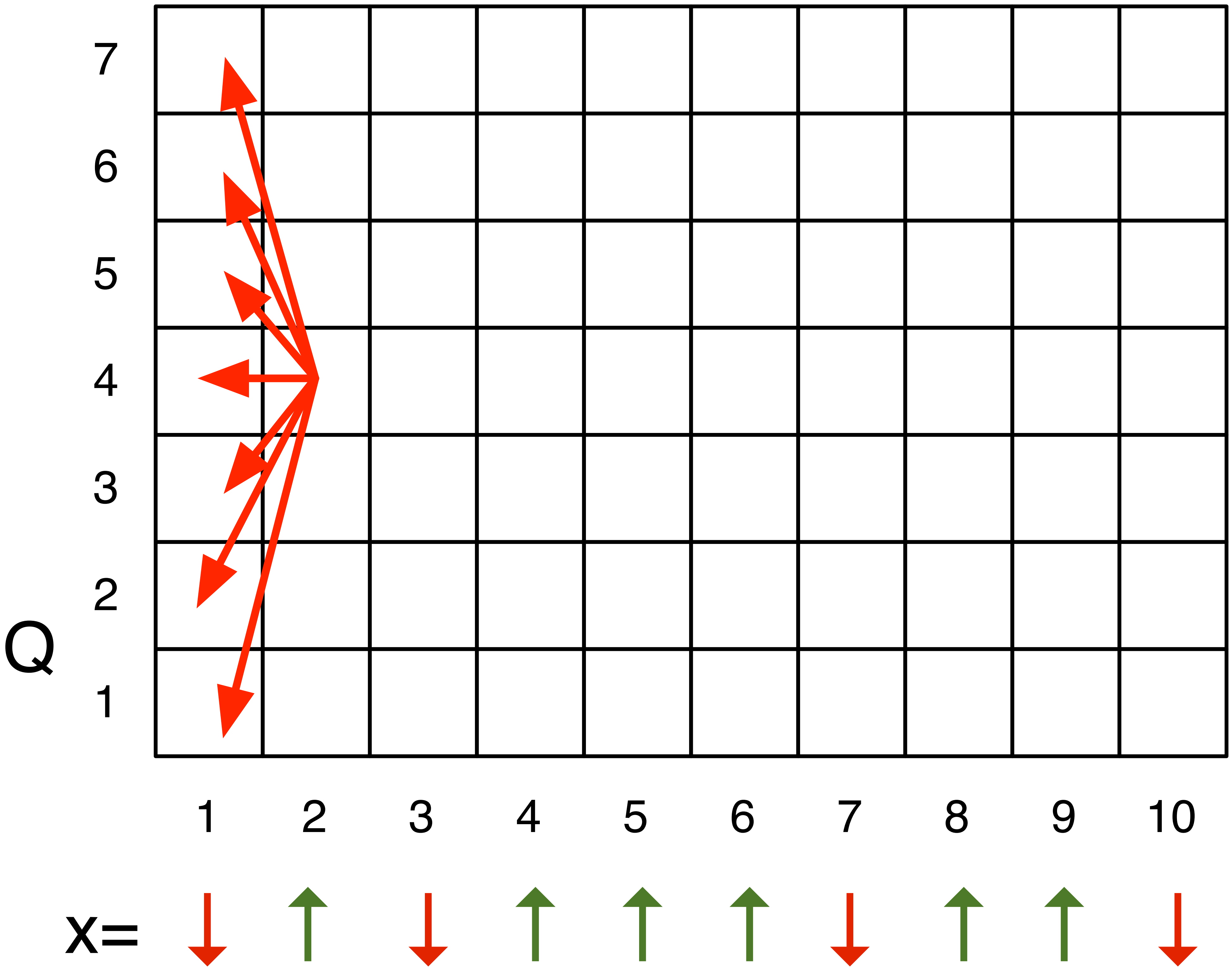
Base case:

$$A[a,1] = \underbrace{\Pr(\pi_1 = a)} \times \underbrace{\Pr(x_1 \mid \pi_1 = a)}$$

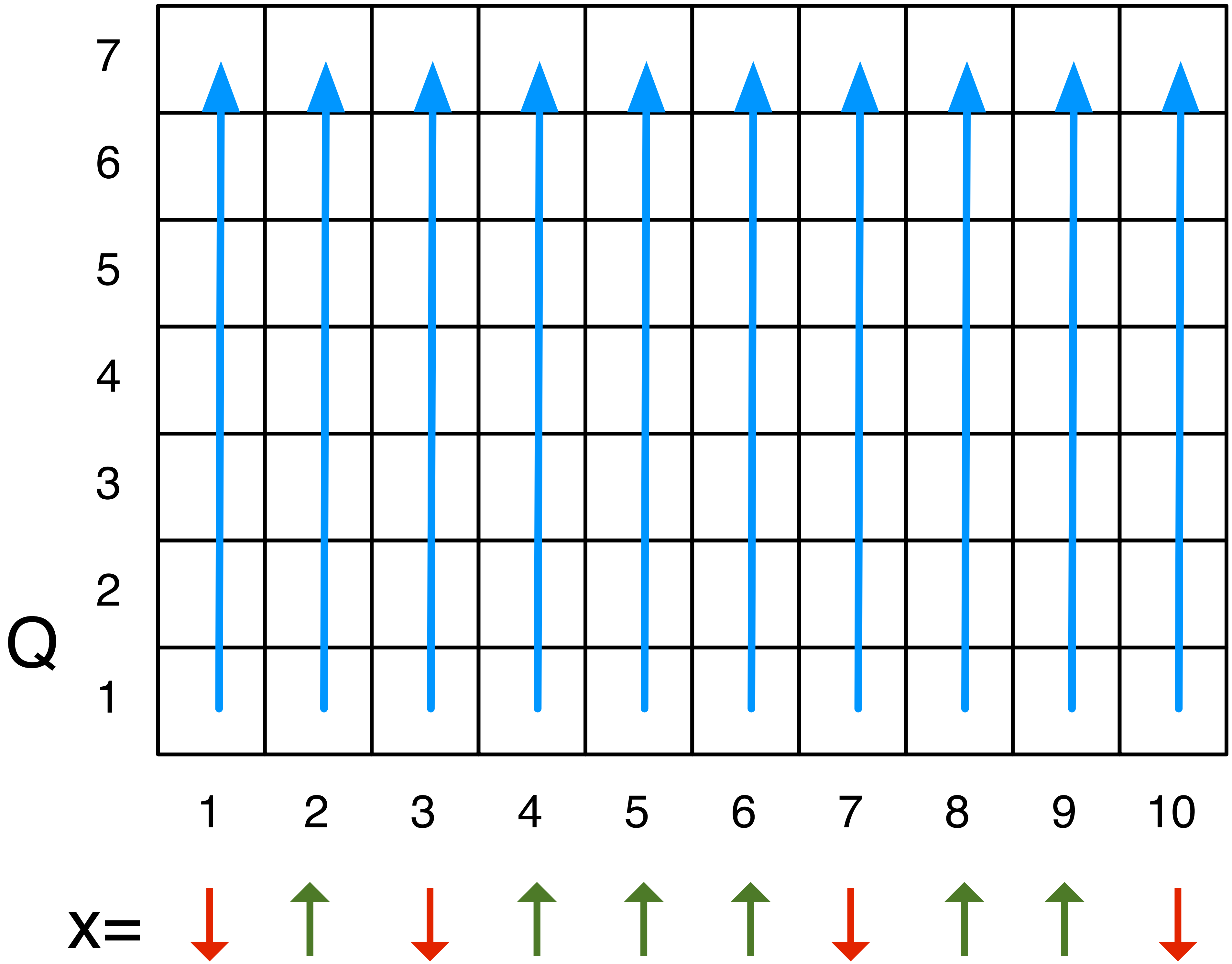
Probability that the first state is *a*

Probability of emitting $x_1$ given the first state is *a*.

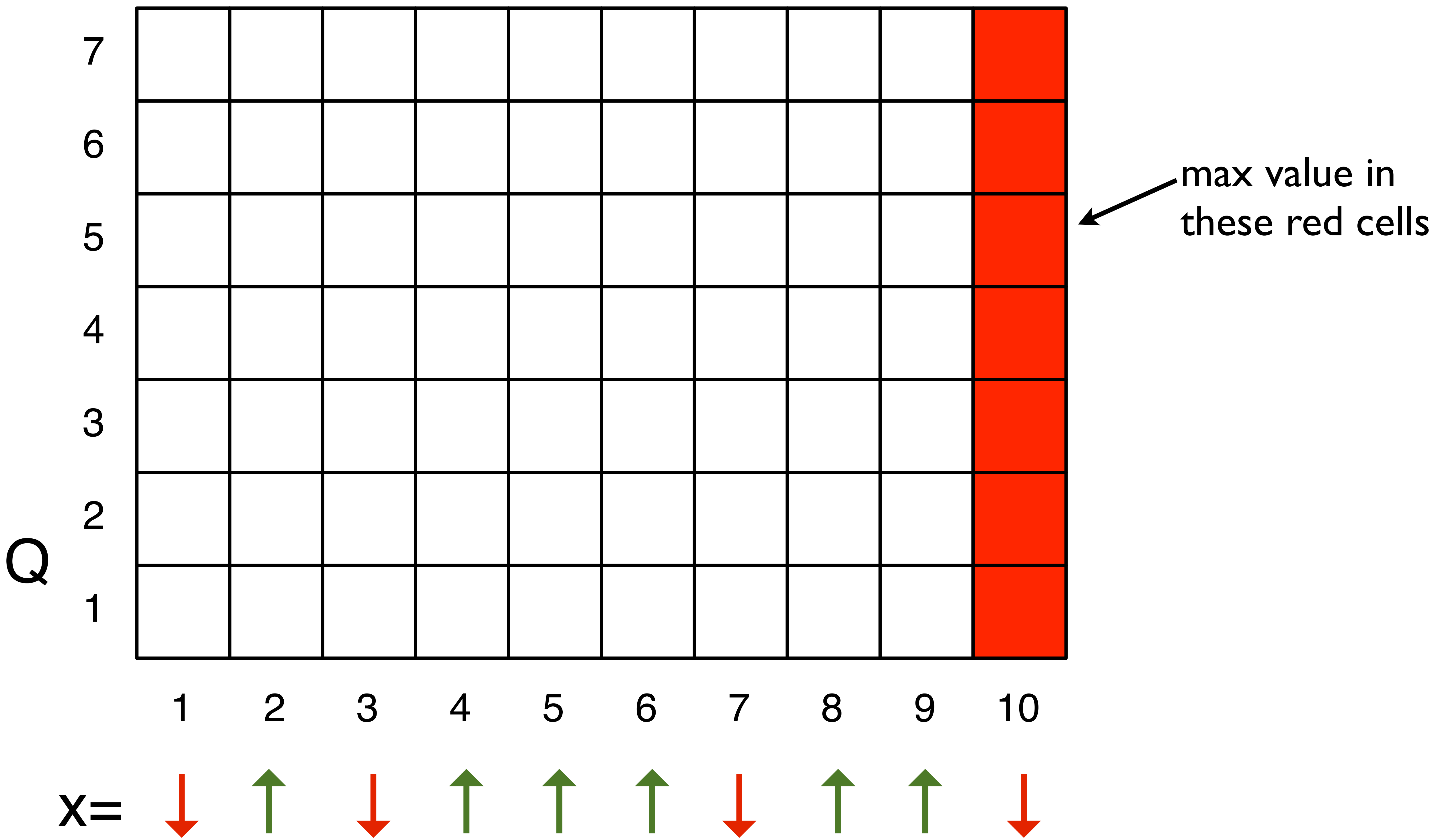
# Which Cells Do We Depend On?



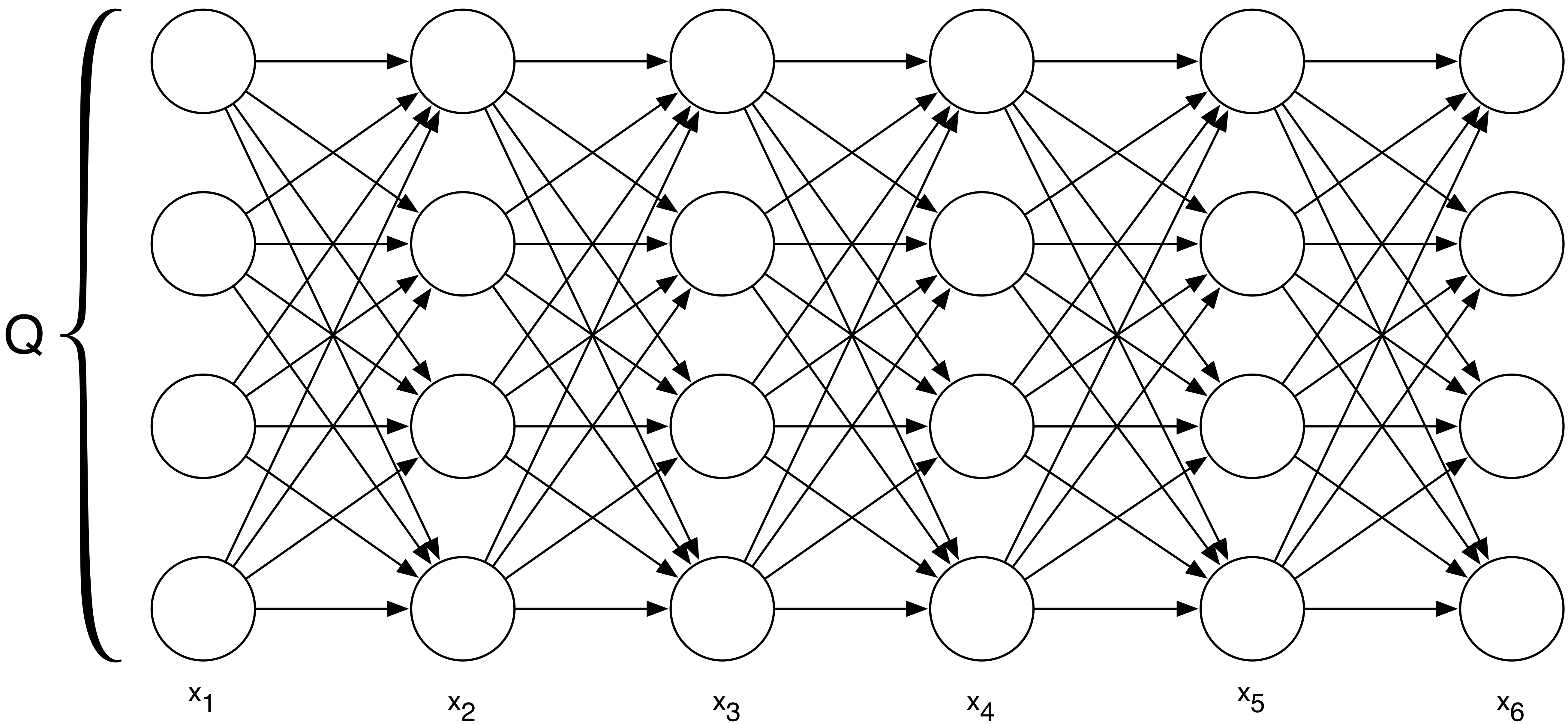
# Order to Fill in the Matrix:



# Where's the answer?



# Graph View of Viterbi



# Running Time

- # of subproblems = $O(n|Q|)$, where n is the length of the sequence.
- Time to solve a subproblem = $O(|Q|)$
- Total running time: $O(n|Q|^2)$

# Using Logs

Typically, we take the log of the probabilities to avoid multiplying a lot of terms:

$$\begin{aligned}\log(A[a,k]) &= \max_{b\in Q}\{\log(A[b,k-1]\times \Pr(b\to a)\times \Pr(x_k \mid \pi_k = a))\}\\ &= \max_{b\in Q}\{\log(A[b,k-1]) + \log(\Pr(b\to a)) + \log(\Pr(x_k \mid \pi_k = a))\}\end{aligned}$$

$$\boxed{\text{Remember: } \log(ab) = \log(a) + \log(b)}$$

Why do we want to avoid multiplying lots of terms?

# Using Logs

Typically, we take the log of the probabilities to avoid multiplying a lot of terms:

$$\begin{aligned}\log(A[a,k]) &= \max_{b\in Q}\{\log(A[b,k-1]\times \Pr(b\to a)\times \Pr(x_k \mid \pi_k = a))\}\\ &= \max_{b\in Q}\{\log(A[b,k-1]) + \log(\Pr(b\to a)) + \log(\Pr(x_k \mid \pi_k = a))\}\end{aligned}$$

Remember: $\log(ab) = \log(a) + \log(b)$

Why do we want to avoid multiplying lots of terms?

Multiplying leads to very small numbers:
0.1 x 0.1 x 0.1 x 0.1 x 0.1 = 0.00001
This can lead to underflow.
Taking logs and adding keeps numbers bigger.

# Estimating HMM Parameters

$$(x^{(1)},\pi^{(1)}) = \begin{matrix} x_1^{(1)} x_2^{(1)} x_3^{(1)} x_4^{(1)} x_5^{(1)} \ldots x_n^{(1)} \\ \pi_1^{(1)} \pi_2^{(1)} \pi_3^{(1)} \pi_4^{(1)} \pi_5^{(1)} \ldots \pi_n^{(1)} \end{matrix}$$

$$(x^{(2)},\pi^{(2)}) = \begin{matrix} x_1^{(2)} x_2^{(2)} x_3^{(2)} x_4^{(2)} x_5^{(2)} \ldots x_n^{(2)} \\ \pi_1^{(2)} \pi_2^{(2)} \pi_3^{(2)} \pi_4^{(2)} \pi_5^{(2)} \ldots \pi_n^{(2)} \end{matrix}$$

Training examples where outputs and paths are known.

# of times transition *a* → *b* is observed.

$$\Pr(a \to b) = \frac{A_{ab}}{\sum_{q \in Q} A_{aq}}$$

# of times *x* was observed to be output from state *a*.

$$\Pr(x \mid a) = \frac{E_{xa}}{\sum_{x \in \Sigma} E_{xq}}$$

# Pseudocounts

# of times transition
$a \to b$ is observed.

$$\Pr(a \to b) = \frac{A_{ab}}{\sum_{q \in Q} A_{aq}}$$

# of times $x$ was
observed to be
output from state $a$.

$$\Pr(x \mid a) = \frac{E_{xa}}{\sum_{x \in \Sigma} E_{xq}}$$

What if a transition or emission is never observed in the training data?
⇒ 0 probability

Meaning that if we observe an example with that transition or emission in the real world, we will give it 0 probability.

But it's unlikely that our training set will be large enough to observe *every* possible transition.

Hence: we take $A_{ab}$ = (#times a → b was observed) + 1 ← "pseudocount"
Similarly for $E_{xa}$.

# Viterbi Training

- **Problem**: typically, in the real would we only have examples of the output x, and we don't know the paths π.

**Viterbi Training Algorithm:**

1. Choose a random set of parameters.
2. Repeat:
   1. Find the best paths.
   2. Use those paths to estimate new parameters.

This is an local search algorithm.

It's also an example of a "Gibbs sampling" style algorithm.

The Baum-Welch algorithm is similar, but doesn't commit to a single best path for each example.

# Some probabilities we are interested in

What is the probability of observing a string *x* under the assumed HMM?

$$\Pr(x) = \sum_{\pi} \Pr(x, \pi)$$

What is the probability of observing *x* using a path where the $i^{th}$ state is *a*?

$$\Pr(x, \pi_i = a) = \sum_{\pi : \pi_i = a} \Pr(x, \pi)$$

What is the probability that the $i^{th}$ state is *a*?

$$\Pr(\pi_i = a | x) = \frac{\Pr(x, \pi_i = a)}{\Pr(x)}$$