Interpretation of entropy as information:
Shannon source coding theorem

surprisal: $-k_B \ln p_n(t)$

entropy: avg surprisal $S(t) = -k_B \sum_n p_n(t) \ln p_n(t)$

imagine scenario: Alice (A) is a prof.
Bob (B) is a grad. stud.

experiment: a system (N=4 states)
(stochastic)

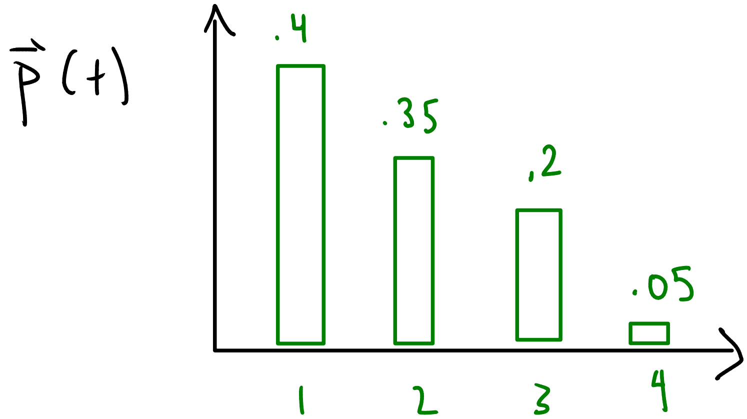
One run:

repeat it many times
- B prepares it in init. dist. $\vec{p}(0)$
- B lets it evolve for time t
- B takes measurement, records state

output: series of measurements:
i.e. 2, 4, 1, 3, 2, ...

A + B know: system trans. matrix W
prob. $\vec{p}(0)$
hence $\vec{p}(t) = W^t \vec{p}(0)$

A doesn't know specific seq. of measurements



case I: Bob is lazy

state	code
1	00
2	01
3	10
4	11

easy to read (every 2 bits is a state)

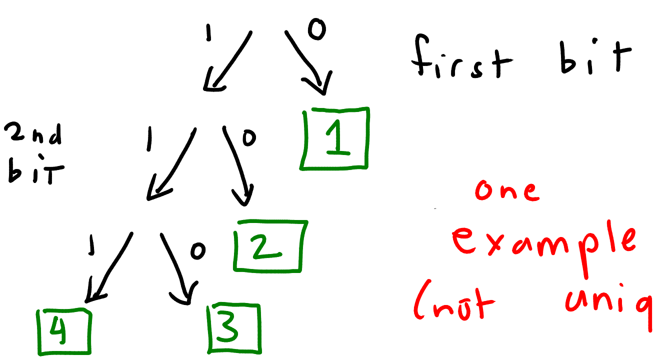
if there were N states in general

mean message length per state

$= \log_2 N$

case II: Bob is clever: a variable length code called a "prefix-free" code: no stop markers b/t states

coding / decoding $\Longleftrightarrow$ binary decision tree



one example (not unique)

state	code	# bits l_n
1	0	1
2	10	2
3	110	3
4	111	3

Seq: 2 1 4 3 $\Rightarrow$ 100111110 ....

avg. # bits per state (cost of sending code)

$$B = \sum_n p_n(t) \ell_n = 0.4 \times 1 + .35 \times 2 + 0.2 \times 3 + .05 \times 3$$

$$= 1.85 \text{ bits / state}$$

for the above code

Shannon proved: among all possible codes (all possible binary decision trees) there is a lower bound:

$$B \geq B_{min} = - \sum_n p_n(t) \log_2 p_n(t)$$

"information entropy"

units: bits

compare to thermo. entropy

$$S(t) = -k_B \sum_n p_n(t) \ln p_n(t)$$

$$B_{min} = \frac{S(t)}{k_B \ln 2}$$

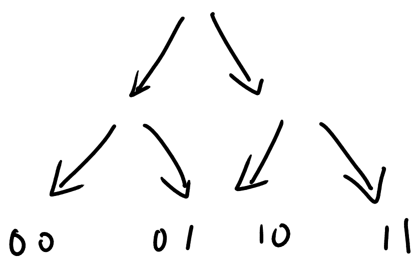
converts from $\frac{J}{K}$ to bits

examples: $p_n(t) = \delta_{n,1}$ all states are 1 in the message

$B_{min} = 0$ (no info needs to be sent)

$$p_n(t) = \frac{1}{4}$$ (all states equally likely in message)

$$B_{min} = 2 \text{ bits}$$ (can't do any better than lazy strategy)



prove $B \geq B_{min} = -\sum_n p_n \log_2 p_n = \frac{S}{k_B \ln 2}$

Consider any code (any binary tree):

fake probability $q_n \equiv 2^{-l_n} = \left(\frac{1}{2}\right)^{l_n}$

= prob. of reaching state n (leaf of a tree) if you start at tree top + choose each fork w/ equal prob.

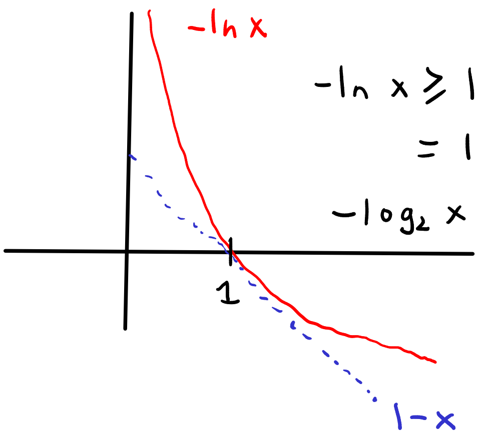
$$\sum_n q_n = 1$$

$$B = \sum_n p_n l_n = -\sum_n p_n \log_2 q_n$$

Shannon's claim $B - B_{min} \geq 0$

$$-\sum_n p_n \log_2 q_n - \left(-\sum_n p_n \log_2 p_n\right)$$

$$= -\sum_n p_n \log_2 \frac{q_n}{p_n} \geq \frac{1}{\ln 2} \sum_n p_n \left(1 - \frac{q_n}{p_n}\right)$$



$-\ln x \geq 1 - x$

$= 1 - x$ iff $x = 1$

$-\log_2 x = \dfrac{-\ln x}{\ln 2}$

$$= \frac{1}{\ln 2}\left(\underbrace{\sum_n p_n}_{1} - \underbrace{\sum_n q_n}_{1}\right)$$

$$= 0$$

$\Rightarrow$ B_{min} is a universal lower bound!

$B = B_{min}$ iff $q_n = p_n$ for each n ($x=1$ for each term)

$2^{-\ell_n} = p_n$

$\Rightarrow$ $\ell_n = -\log_2 p_n$ in an optimal code

note: $-\log_2 p_n$ is not always integer so in general we can approach but not reach B_{min}