Information theory is the study of quantification of discrete (or discretized) data. It is used in a wide variety of fields, including but not limited to:

- ZIP data compression
- AI game strategies
- Error correction

Here we will use a very small slice.

**Def** Let \((\Omega, F, P)\) be a measurable space (something that behaves nicely).

with \(P(E)\) being the probability of some event \(E \in F\) and \(P(\Omega) = 1\). Then \((\Omega, F, P)\) is a probability space.

Implicitly, the following axioms hold.

1) \(\forall E \in F, \ P(E) \in \mathbb{R}_{\geq 0}\) (note that this implies probabilities are finite)

2) \(P(\Omega) = 1\) (can be infinitely \(\infty\))

3) Let \(\{E_i\}_{i=1}^{\infty}\) be disjoint and countable. Then they satisfy

\[
P\left(\bigcup_{i} E_i\right) = \sum_{i} P(E_i)
\]

We are going to use discrete probability spaces, that is, ones where \(\Omega\) is countable.
The standard unit of information is the bit. A bit of information is something that narrows down the sample space by half.

Ex) Let $p$ be the uniform probability on $\mathbb{Z}_4$. If we want to know a number $n \in \mathbb{Z}_4$, and we know $n \geq 8$, then we know that half of $\mathbb{Z}_4$ (0, 1, ..., 6, 7) are not $n$, hence we have 1 bit of information. If we also know that $n \leq 12$, then we also know that $n \neq 12$, 13, 14/15. $n$ can only have 4 of 16 possible values then, so we've halved the sample space twice and have 2 bits of information. If we instead knew $n \geq 4$ or $n \leq 42$, we eliminate no further possibilities (from $n \geq 8$) and thus gain no information from those facts.

Def) The information of some event $E \in F$ (a restriction of the set of possibilities from the sample space) is given by

$$I(E) = -\log p(E)$$

For the uninitiated, we obtain this as follows.

$$\left\{ \begin{align*}
\left( \frac{1}{2} \right)^{I(E)} &= p(E) \\
\Rightarrow \frac{1}{2}^{I(E)} &= \frac{1}{p(E)} \\
\Rightarrow I(E) &= \log \left( \frac{1}{p(E)} \right) \\
\Rightarrow I(E) &= -\log p(E)
\end{align*} \right.$$
Ex) Suppose our sample space is the alphabet (isomorphic to \( \mathbb{Z}_{26} \)). We want to guess a 5-letter word. To simplify matters, we'll assume any combination of 5 letters is equally likely (even combinations which are not words) rather than look up usage data (how often we use each word).

We got the word from a fading book and know the second letter is an e and the last letter is an n. How much information do we have?

\[
P(2^{nd} \text{ letter } = e \land 5^{th} \text{ letter } = n) = \frac{1}{26} \cdot \frac{1}{26} = \frac{1}{676}
\]

\[
I(\text{"e"}) = -\log \frac{1}{676} \approx 9.4 \text{ bits}
\]

How many bits of information do we need to identify the word?

\[
I(\text{the word}) = -\log(\frac{1}{26})^5 = -5 \log(\frac{1}{26}) \approx 23.5 \text{ bits}
\]

Notice that information likes to add the way probabilities like to add.

Ex) Suppose we have a non-uniform probability on \( \mathbb{Z}_4 \), that is

\[
P(n) = \begin{cases} 
\frac{1}{2} & n = 0 \\
\frac{1}{4} & n = 1 \\
\frac{1}{8} & n = 2 \text{ or } 3 
\end{cases}
\]

\[
\text{If } n = 1, \text{ how much information do we have?} \quad 1 \text{ bit}
\]

\[
\text{If } n > 2, \quad \text{"} \quad 2 \text{ bits}
\]

\[
\text{If } n \text{ is not } 2, \quad \text{"} \quad \approx 0.42 \text{ bits}
\]

Notice that knowing \( n = 1 \) tells us very little. If \( P \) were uniform, \( n = 1 \) would give us 2 bits of information. We could write a compression algorithm for this!

For those interested in more, look into entropy (aka expected information).
If we want to sort an array A of n elements, how much information do we need if we know nothing a priori about A? A has n! possible permutations, so if each is equally likely, \[ I(A) = -\log\frac{1}{n!} = \log(n!) \approx n \log n \text{ bits.} \]

Note that this in and of itself does not imply the best sorting algorithm runs in \( \Theta(n \log n) \) time. Each observation we make can contain either more or less than 1 bit.

If we make a comparison, however, we need to determine how much information we obtain. Since some sorts run in \( \Theta(n) \) and some in \( \Omega(n^2) \), it's clearly not necessarily 1 bit per comparison.

To formalize the notion of a "comparison", we define it to be a yes/no query to an oracle about A. We want to know the minimum number of queries \( k \) so that we can guess A with probability 1 (the state space is discrete, so probability 1 means we know A).

Skipping the intermediary setup, we invoke Fano’s inequality to get that \( P_e \) (the probability of error) satisfies

\[ P_e \leq 1 - \frac{k+1}{n \log n}, \]

To get zero error, we set \( P_e = 0 \) and obtain

\[ 1 \leq \frac{k+1}{n \log n}, \]

hence \( k \geq n \log n - 1 \) and the best we can do (on average) is \( \Theta(n \log n) \). Remember, this is a probabilistic argument, so we're dealing with average case. This in turn implies the WC.