Entropy, Randomness, and Information

Lecture 26
December 3, 2018
26.1: Entropy
“If only once - only once - no matter where, no matter before what audience - I could better the record of the great Rastelli and juggle with thirteen balls, instead of my usual twelve, I would feel that I had truly accomplished something for my country. But I am not getting any younger, and although I am still at the peak of my powers there are moments - why deny it? - when I begin to doubt - and there is a time limit on all of us.”

Entropy: Definition

Definition

The \textit{entropy} in bits of a discrete random variable \( X \) is

\[ H(X) = - \sum_x \Pr[X = x] \lg \Pr[X = x]. \]

Equivalently, \( H(X) = \mathbb{E} \left[ \lg \frac{1}{\Pr[X]} \right] \).
Entropy
Clicker question

Consider $X$ a random variable that picks its value uniformly from $1, \ldots, n$. We have that its entropy $\mathbb{H}(X) = - \sum_x \Pr[X = x] \log \Pr[X = x]$ is

1. $O(\log n)$.
2. $O(n)$.
3. $\ln n$.
4. $n \times \ln n$.
5. $\lg n$. 
Entropy intuition...

Intuition...
\( \mathbb{H}(X) \) is the number of \textit{fair} coin flips that one gets when getting the value of \( X \).

Interpretation from last lecture...
Consider a (huge) string \( S = s_1s_2 \ldots s_n \) formed by picking characters independently according to \( X \). Then

\[
|S| \mathbb{H}(X) = n \mathbb{H}(X)
\]

is the minimum number of bits one needs to store the string \( S \) (when we compress it).
Entropy II

Clicker question

Consider $X$ a random variable that

$$\Pr[X = i] = \frac{1/i}{\alpha},$$

for $i = 1, \ldots, \infty$, where $\alpha = \sum_{i=1}^{\infty} 1/i$.

The entropy of $X$ is

$$\mathbb{H}(X) = -\sum_x \Pr[X = x] \lg \Pr[X = x]$$

equal to

1. $O(1)$.
2. $O(n)$.
3. 0.
4. $\infty$. 

Consider $X$ a random variable that

$$\Pr[X = i] = \frac{1/i^2}{\alpha},$$

for $i = 2, \ldots, \infty$, where $\alpha = \sum_{i=2}^{\infty} 1/i^2$. The entropy of $X$ is

$$\mathbb{H}(X) = - \sum_x \Pr[X = x] \log \Pr[X = x]$$

equal to

1. $O(1)$.
2. $O(n)$.
3. 0.
4. $\infty$. 
Entropy V

Clicker question

Consider $X$ a random variable that

$$\Pr[X = i] = 2^{-i}$$

for $i = 1, \ldots, \infty$. The entropy of $X$ is

$$\mathbb{H}(X) = - \sum_x \Pr[X = x] \lg \Pr[X = x]$$

equal to

1. $O(1)$.
2. $O(n)$.
3. 0.
4. $\infty$.
5. $\lg n$. 
Entropy of a geometric distribution...

\[ H(X) = - \sum_x \Pr(X = x) \lg \Pr(X = x) \]

\[ = - \sum_{i=1}^{\infty} \frac{1}{2^i} \lg \frac{1}{2^i} \]

\[ = \sum_{i=1}^{\infty} \frac{1}{2^i} \lg 2^i \]

\[ = \sum_{i=1}^{\infty} \frac{i}{2^i} \]

\[ = 2. \]
Entropy of a geometric distribution...

\[ \mathbb{H}(X) = - \sum_{x} \Pr[X = x] \log \Pr[X = x] \]

\[ = - \sum_{i=1}^{\infty} \frac{1}{2^i} \log \frac{1}{2^i} \]

\[ = \sum_{i=1}^{\infty} \frac{1}{2^i} \log 2^i \]

\[ = \sum_{i=1}^{\infty} \frac{i}{2^i} \]

\[ = 2. \]
Entropy of a geometric distribution...

\[ H(X) = - \sum_{x} \Pr[X = x] \log \Pr[X = x] \]

\[ = - \sum_{i=1}^{\infty} \frac{1}{2^i} \log \frac{1}{2^i} \]

\[ = \sum_{i=1}^{\infty} \frac{1}{2^i} \log 2^i \]

\[ = \sum_{i=1}^{\infty} \frac{i}{2^i} \]

\[ = 2. \]
Entropy of a geometric distribution...

\[ H(X) = - \sum_{x} \Pr[X = x] \log \Pr[X = x] \]

\[ = - \sum_{i=1}^{\infty} \frac{1}{2^i} \log \frac{1}{2^i} \]

\[ = \sum_{i=1}^{\infty} \frac{i}{2^i} \log 2^i \]

\[ = 2. \]
Entropy of a geometric distribution...

\[ H(X) = - \sum_x \Pr[X = x] \, \lg \Pr[X = x] \]

\[ = - \sum_{i=1}^{\infty} \frac{1}{2^i} \, \lg \frac{1}{2^i} \]

\[ = \sum_{i=1}^{\infty} \frac{i}{2^i} \, \lg 2^i \]

\[ = \sum_{i=1}^{\infty} \frac{i}{2^i} \]

\[ = 2. \]
Binary entropy

\[ H(X) = - \sum_x \Pr[X = x] \log \Pr[X = x] \]

Definition

The binary entropy function \( H(p) \) for a random binary variable that is \( 1 \) with probability \( p \), is

\[ H(p) = -p \log p - (1 - p) \log(1 - p). \]

We define \( H(0) = H(1) = 0 \).

Q: How many truly random bits are there when given the result of flipping a single coin with probability \( p \) for heads?
Binary entropy

\[ H(X) = - \sum_x \Pr[X = x] \log \Pr[X = x] \]

**Definition**

The *binary entropy* function \( H(p) \) for a random binary variable that is 1 with probability \( p \), is

\[ H(p) = -p \log p - (1 - p) \log(1 - p) \]

We define \( H(0) = H(1) = 0 \).

Q: How many truly random bits are there when given the result of flipping a single coin with probability \( p \) for heads?
Binary entropy

\[ H(X) = - \sum_x \Pr[X = x] \log \Pr[X = x] \]

\[ \Rightarrow \]

Definition

The \textit{binary entropy} function \( H(p) \) for a random binary variable that is 1 with probability \( p \), is

\[ H(p) = -p \log p - (1-p) \log(1-p). \]

We define \( H(0) = H(1) = 0 \).

Q: How many truly random bits are there when given the result of flipping a single coin with probability \( p \) for heads?
Binary entropy:
\[ H(p) = -p \log_2 p - (1 - p) \log_2(1 - p) \]

1. \( H(p) \) is a concave symmetric around \( 1/2 \) on the interval \([0, 1]\).
2. maximum at \( 1/2 \).
3. \( H(3/4) \approx 0.8113 \) and \( H(7/8) \approx 0.5436 \).
4. \( \implies \) coin that has \( 3/4 \) probably to be heads have higher amount of “randomness” in it than a coin
Binary entropy:

\[ \mathbb{H}(p) = -p \log_2 p - (1 - p) \log_2 (1 - p) \]

1. \( \mathbb{H}(p) \) is a concave symmetric around \( 1/2 \) on the interval \([0, 1]\).
2. maximum at \( 1/2 \).
3. \( \mathbb{H}(3/4) \approx 0.8113 \) and \( \mathbb{H}(7/8) \approx 0.5436 \).
4. coin that has \( 3/4 \) probably to be heads have higher amount of “randomness” in it than a coin.
Binary entropy:

$$H(p) = -p \lg p - (1-p) \lg(1-p)$$

1. $H(p)$ is a concave symmetric around $1/2$ on the interval $[0, 1]$.
2. maximum at $1/2$.
3. $H(3/4) \approx 0.8113$ and $H(7/8) \approx 0.5436$.
4. coin that has $3/4$ probably to be heads have higher amount of “randomness” in it than a coin
Binary entropy:
\[ H(p) = -p \log p - (1 - p) \log(1 - p) \]

1. \( H(p) \) is a concave symmetric around \( 1/2 \) on the interval \([0, 1]\).
2. maximum at \( 1/2 \).
3. \( H(3/4) \approx 0.8113 \) and \( H(7/8) \approx 0.5436 \).
4. \( \implies \) coin that has \( 3/4 \) probably to be heads have higher amount of “randomness” in it than a coin.
And now for some unnecessary math

1. \( H(p) = -p \log p - (1 - p) \log(1 - p) \)
2. \( H'(p) = -\log p + \log(1 - p) = \log \frac{1-p}{p} \)
3. \( H''(p) = \frac{p}{1-p} \cdot \left( -\frac{1}{p^2} \right) = -\frac{1}{p(1-p)} \).
4. \( \Rightarrow H''(p) \leq 0 \), for all \( p \in (0, 1) \), and the \( H(\cdot) \) is concave.
5. \( H'(1/2) = 0 \Rightarrow H(1/2) = 1 \max \) of binary entropy.
6. \( \Rightarrow \) balanced coin has the largest amount of randomness in it.
And now for some unnecessary math

1. $\mathbb{H}(p) = -p \lg p - (1 - p) \lg(1 - p)$

2. $\mathbb{H}'(p) = -\lg p + \lg(1 - p) = \lg \frac{1-p}{p}$

3. $\mathbb{H}''(p) = \frac{p}{1-p} \cdot \left( -\frac{1}{p^2} \right) = -\frac{1}{p(1-p)}$.

4. $\implies \mathbb{H}''(p) \leq 0$, for all $p \in (0, 1)$, and the $\mathbb{H}(\cdot)$ is concave.

5. $\mathbb{H}'(1/2) = 0 \implies \mathbb{H}(1/2) = 1$ max of binary entropy.

6. $\implies$ balanced coin has the largest amount of randomness in it.
And now for some unnecessary math

1. \( H(p) = -p \log p - (1 - p) \log (1 - p) \)
2. \( H'(p) = - \log p + \log (1 - p) = \log \frac{1-p}{p} \)
3. \( H''(p) = \frac{p}{1-p} \cdot \left(-\frac{1}{p^2}\right) = -\frac{1}{p(1-p)} \).
4. \( \Rightarrow H''(p) \leq 0 \), for all \( p \in (0, 1) \), and the \( H(\cdot) \) is concave.
5. \( H'(1/2) = 0 \Rightarrow H(1/2) = 1 \) max of binary entropy.
6. \( \Rightarrow \) balanced coin has the largest amount of randomness in it.
And now for some unnecessary math

1. $$\mathbb{H}(p) = -p \log p - (1 - p) \log(1 - p)$$
2. $$\mathbb{H}'(p) = -\log p + \log(1 - p) = \log \frac{1-p}{p}$$
3. $$\mathbb{H}''(p) = \frac{p}{1-p} \cdot \left(-\frac{1}{p^2}\right) = -\frac{1}{p(1-p)}.$$  
4. $$\implies \mathbb{H}''(p) \leq 0,$$ for all $$p \in (0, 1),$$ and the $$\mathbb{H}(\cdot)$$ is concave.
5. $$\mathbb{H}'(1/2) = 0 \implies \mathbb{H}(1/2) = 1$$ max of binary entropy.
6. $$\implies$$ balanced coin has the largest amount of randomness in it.
And now for some unnecessary math

1. \( H(p) = -p \log p - (1 - p) \log(1 - p) \)

2. \( H'(p) = -\log p + \log(1 - p) = \log \frac{1-p}{p} \)

3. \( H''(p) = \frac{p}{1-p} \cdot \left(-\frac{1}{p^2}\right) = -\frac{1}{p(1-p)} \).

4. \( \implies H''(p) \leq 0 \), for all \( p \in (0, 1) \), and the \( H(\cdot) \) is concave.

5. \( H'(1/2) = 0 \implies H(1/2) = 1 \) max of binary entropy.

6. \( \implies \) balanced coin has the largest amount of randomness in it.
And now for some unnecessary math

1. $H(p) = -p \log p - (1 - p) \log(1 - p)$
2. $H'(p) = - \log p + \log(1 - p) = \log \frac{1 - p}{p}$
3. $H''(p) = \frac{p}{1 - p} \cdot \left( - \frac{1}{p^2} \right) = - \frac{1}{p(1 - p)}$
4. $\implies H''(p) \leq 0$, for all $p \in (0, 1)$, and the $H(\cdot)$ is concave.
5. $H'(1/2) = 0 \implies H(1/2) = 1$ max of binary entropy.
6. $\implies$ balanced coin has the largest amount of randomness in it.
26.3: Squeezing randomness
Task at hand: Squeezing good random bits...
...out of bad random bits...

1. \( b_1, \ldots, b_n \): result of \( n \) coin flips...
2. From a faulty coin!
3. \( p \): probability for head.
4. We need fair bit coins!
5. Convert \( b_1, \ldots, b_n \rightarrow b'_1, \ldots, b'_m \).
6. New bits must be truly random: Probability for head is \( 1/2 \).
7. Q: How many truly random bits can we extract?
Task at hand: Squeezing good random bits...

...out of bad random bits...

1. \( b_1, \ldots, b_n \): result of \( n \) coin flips...
2. From a faulty coin!
3. \( p \): probability for head.
4. We need fair bit coins!
5. Convert \( b_1, \ldots, b_n \rightarrow b'_1, \ldots, b'_m \).
6. New bits must be truly random: Probability for head is \( 1/2 \).
7. \( Q \): How many truly random bits can we extract?
Task at hand: Squeezing good random bits... 

...out of bad random bits...

1. $b_1, \ldots, b_n$: result of $n$ coin flips...
2. From a faulty coin!
3. $p$: probability for head.
4. We need fair bit coins!
5. Convert $b_1, \ldots, b_n \implies b'_1, \ldots, b'_m$.
6. New bits must be truly random: Probability for head is $1/2$.
7. Q: How many truly random bits can we extract?
Task at hand: Squeezing good random bits...
...out of bad random bits...

1. \( b_1, \ldots, b_n \): result of \( n \) coin flips...
2. From a faulty coin!
3. \( p \): probability for head.
4. We need fair bit coins!
5. Convert \( b_1, \ldots, b_n \Rightarrow b'_1, \ldots, b'_m \).
6. New bits must be truly random: Probability for head is \( 1/2 \).
7. Q: How many truly random bits can we extract?
Task at hand: Squeezing good random bits...

...out of bad random bits...

1. \(b_1, \ldots, b_n\): result of \(n\) coin flips...
2. From a faulty coin!
3. \(p\): probability for head.
4. We need fair bit coins!
5. Convert \(b_1, \ldots, b_n \implies b'_1, \ldots, b'_m\).
6. New bits must be truly random: Probability for head is \(1/2\).
7. Q: How many truly random bits can we extract?
Task at hand: Squeezing good random bits...
...out of bad random bits...

1. $b_1, \ldots, b_n$: result of $n$ coin flips...
2. From a faulty coin!
3. $p$: probability for head.
4. We need fair bit coins!
5. Convert $b_1, \ldots, b_n \implies b'_1, \ldots, b'_m$.
6. **New bits must be truly random**: Probability for head is $1/2$.
7. **Q**: How many truly random bits can we extract?
Task at hand: Squeezing good random bits...
...out of bad random bits...

1. $b_1, \ldots, b_n$: result of $n$ coin flips...
2. From a faulty coin!
3. $p$: probability for head.
4. We need fair bit coins!
5. Convert $b_1, \ldots, b_n \implies b'_1, \ldots, b'_m$.
6. **New bits must be truly random**: Probability for head is $1/2$.
7. **Q**: How many truly random bits can we extract?
Intuitively...
Squeezing good random bits out of bad random bits...

Question...
Given the result of $n$ coin flips: $b_1, \ldots, b_n$ from a faulty coin, with head with probability $p$, how many truly random bits can we extract?

If believe intuition about entropy, then this number should be $\approx n \mathbb{H}(p)$. 
Back to Entropy

1. **entropy** of $X$ is
   \[
   H(X) = - \sum_x \Pr[X = x] \log \Pr[X = x].
   \]

2. Entropy of uniform variable.

Example

A random variable $X$ that has probability $1/n$ to be $i$, for $i = 1, \ldots, n$, has entropy

\[
H(X) = - \sum_{i=1}^{n} \frac{1}{n} \log \frac{1}{n} = \log n.
\]

3. Entropy is oblivious to the exact values random variable can have.

4. Random variables over $-1, +1$ with equal probability has the same entropy (i.e., 1) as a fair coin.
Back to Entropy

1. **Entropy** of $X$ is
   \[ H(X) = - \sum_x \Pr[X = x] \log \Pr[X = x]. \]

2. Entropy of uniform variable.

### Example

A random variable $X$ that has probability $1/n$ to be $i$, for $i = 1, \ldots, n$, has entropy
\[ H(X) = - \sum_{i=1}^n \frac{1}{n} \log \frac{1}{n} = \log n. \]

3. Entropy is oblivious to the exact values random variable can have.

4. Random variables over $-1, +1$ with equal probability has the same entropy (i.e., $1$) as a fair coin.
Back to Entropy

1. **entropy** of $X$ is
   \[ H(X) = - \sum_x \Pr[X = x] \lg \Pr[X = x]. \]

2. Entropy of uniform variable.

**Example**

A random variable $X$ that has probability $1/n$ to be $i$, for $i = 1, \ldots, n$, has entropy
\[ H(X) = - \sum_{i=1}^n \frac{1}{n} \lg \frac{1}{n} = \lg n. \]

3. Entropy is oblivious to the exact values random variable can have.

4. Random variables over $-1, +1$ with equal probability has the same entropy (i.e., $1$) as a fair coin.
Back to Entropy

1. **entropy** of $X$ is
   \[ H(X) = - \sum_x \Pr[X = x] \log \Pr[X = x]. \]

2. Entropy of uniform variable.

**Example**

A random variable $X$ that has probability $1/n$ to be $i$, for $i = 1, \ldots, n$, has entropy
\[ H(X) = - \sum_{i=1}^{n} \frac{1}{n} \log \frac{1}{n} = \log n. \]

3. Entropy is oblivious to the exact values random variable can have.

4. Random variables over $-1, +1$ with equal probability has the same entropy (i.e., $1$) as a fair coin.
Back to Entropy

1. **entropy** of $X$ is

   \[ H(X) = - \sum_x \Pr[X = x] \log \Pr[X = x]. \]

2. Entropy of uniform variable.

**Example**

A random variable $X$ that has probability $1/n$ to be $i$, for $i = 1, \ldots, n$, has entropy

\[ H(X) = - \sum_{i=1}^{n} \frac{1}{n} \log \frac{1}{n} = \log n. \]

3. Entropy is oblivious to the exact values random variable can have.

4. \(\implies\) random variables over \(-1, +1\) with equal probability has the same entropy (i.e., 1) as a fair coin.
Flipper
Clicker question

You are given a coin that is head with probability $p$, and tail with probability $q = 1 - p$. We flip it three times, and get the string $S = s_1 s_2 s_3$. We have the following:

1. $\Pr[S = 001] = \Pr[S = 011] = pq^2$.
2. $\Pr[S = 101] = \Pr[S = 110] = \Pr[S = 011] = pq^2$.
3. $\Pr[S = 111] = \Pr[S = 000] = q^3$.
4. $\Pr[S = 001] = \Pr[S = 010] = \Pr[S = 100] = pq^2$.
5. $\Pr[S = 000] + \Pr[S = 111] = (p + q)^3$. 
Lemma: Entropy additive for independent variables

Lemma

Let $X$ and $Y$ be two independent random variables, and let $Z$ be the random variable $(X, Y)$. Then

$$H(Z) = H(X) + H(Y).$$
Proof

In the following, summation are over all possible values that the variables can have. By the independence of $X$ and $Y$ we have

$$\mathbb{H}(Z) = \sum_{x,y} \Pr[(X, Y) = (x, y)] \lg \frac{1}{\Pr[(X, Y) = (x, y)]}$$

$$= \sum_{x,y} \Pr[X = x] \Pr[Y = y] \lg \frac{1}{\Pr[X = x] \Pr[Y = y]}$$

$$= \sum_{x} \sum_{y} \Pr[X = x] \Pr[Y = y] \lg \frac{1}{\Pr[X = x]}$$

$$+ \sum_{y} \sum_{x} \Pr[X = x] \Pr[Y = y] \lg \frac{1}{\Pr[Y = y]}$$
Proof continued

\[ \mathbb{H}(Z) = \sum_x \sum_y \Pr[X = x] \Pr[Y = y] \log \frac{1}{\Pr[X = x]} \]

\[ + \sum_y \sum_x \Pr[X = x] \Pr[Y = y] \log \frac{1}{\Pr[Y = y]} \]

\[ = \sum_x \Pr[X = x] \log \frac{1}{\Pr[X = x]} \]

\[ + \sum_y \Pr[Y = y] \log \frac{1}{\Pr[Y = y]} \]

\[ = \mathbb{H}(X) + \mathbb{H}(Y). \]
The entropy of \( Y \)...

Clicker question

Consider a binary string \( Y \) generated by flipping a coin \( n \) times, where the probability for heads is \( p \). Then we have that

1. \( \mathbb{H}(Y) = \ln \left( \frac{n}{np} \right) \).
2. \( \mathbb{H}(Y) = np \).
3. \( \mathbb{H}(Y) = n \mathbb{H}(p) \).
4. \( \mathbb{H}(Y) = n - n \mathbb{H}(p) \).
5. \( \mathbb{H}(Y) = \mathbb{H}(np) \).
Bounding the binomial coefficient using entropy

Lemma

\( q \in [0, 1] \)

\( nq \) is integer in the range \([0, n]\).

Then

\[
\frac{2^{nH(q)}}{n + 1} \leq \binom{n}{nq} \leq 2^{nH(q)}.
\]
Proof

Holds if $q = 0$ or $q = 1$, so assume $0 < q < 1$. We have

\[
\binom{n}{nq} q^{nq} (1 - q)^{n - nq} \leq (q + (1 - q))^n = 1.
\]

We also have:

\[
q^{-nq} (1 - q)^{-(1-q)n} = 2^n \left( -q \lg q - (1-q) \lg(1-q) \right) = 2^n \mathbb{H}(q), \text{ we have}
\]

\[
\binom{n}{nq} \leq q^{-nq} (1 - q)^{-(1-q)n} = 2^n \mathbb{H}(q).
\]
Proof

Holds if $q = 0$ or $q = 1$, so assume $0 < q < 1$. We have

$$\binom{n}{nq} q^{nq} (1 - q)^{n-nq} \leq (q + (1 - q))^n = 1.$$ 

We also have:

$$q^{-nq} (1 - q)^{- (1-q)n} = 2^n \left(-q \log q - (1-q) \log(1-q)\right) = 2^n \mathbb{H}(q),$$

so

$$\binom{n}{nq} \leq q^{-nq} (1 - q)^{- (1-q)n} = 2^n \mathbb{H}(q).$$
Proof

Holds if $q = 0$ or $q = 1$, so assume $0 < q < 1$. We have

$$\binom{n}{nq} q^{nq} (1 - q)^{n - nq} \leq (q + (1 - q))^n = 1.$$  

We also have: $q^{-nq} (1 - q)^{-(1-q)n} = 2^n \left(-q \log q - (1-q) \log(1-q)\right) = 2^n \mathcal{H}(q)$, we have

$$\binom{n}{nq} \leq q^{-nq} (1 - q)^{-(1-q)n} = 2^n \mathcal{H}(q).$$
Proof

Holds if $q = 0$ or $q = 1$, so assume $0 < q < 1$. We have

$$\binom{n}{nq} q^{nq} (1 - q)^{n - nq} \leq (q + (1 - q))^n = 1.$$ 

We also have:

$$q^{-nq} (1 - q)^{(1-q)n} = 2^n (-q \log q - (1-q) \log (1-q)) = 2^n H(q),$$

we have

$$\binom{n}{nq} \leq q^{-nq} (1 - q)^{(1-q)n} = 2^n H(q).$$
**Proof continued**

Other direction...

1. \( \mu(k) = \binom{n}{k} q^k (1 - q)^{n-k} \)

2. \( \sum_{i=0}^{n} \binom{n}{i} q^i (1 - q)^{n-i} = \sum_{i=0}^{n} \mu(i). \)

3. Claim: \( \mu(nq) = \binom{n}{nq} q^{nq} (1 - q)^{n-nq} \)

   largest term in \( \sum_{k=0}^{n} \mu(k) = 1. \)

4. \( \Delta_k = \mu(k) - \mu(k + 1) = \)

   \( \binom{n}{k} q^k (1 - q)^{n-k} \left( 1 - \frac{n-k}{k+1} \frac{q}{1-q} \right) , \)

5. sign of \( \Delta_k = \) size of last term...

6. \( \text{sign}(\Delta_k) = \text{sign} \left( 1 - \frac{(n-k)q}{(k+1)(1-q)} \right) \)

   \( = \text{sign} \left( \frac{(k+1)(1-q)-(n-k)q}{(k+1)(1-q)} \right) . \)
Proof continued

Other direction...

1. \( \mu(k) = \binom{n}{k} q^k (1 - q)^{n-k} \)

2. \( \sum_{i=0}^{n} \binom{n}{i} q^i (1 - q)^{n-i} = \sum_{i=0}^{n} \mu(i). \)

3. Claim: \( \mu(nq) = \binom{n}{nq} q^{nq} (1 - q)^{n-nq} \) largest term in \( \sum_{k=0}^{n} \mu(k) = 1. \)

4. \( \Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1 - q)^{n-k} \left( 1 - \frac{n-k}{k+1} \frac{q}{1-q} \right), \)

5. sign of \( \Delta_k \) = size of last term...

6. \( \text{sign}(\Delta_k) = \text{sign} \left( 1 - \frac{(n-k)q}{(k+1)(1-q)} \right) \)
   \[ = \text{sign} \left( \frac{(k+1)(1-q)-(n-k)q}{(k+1)(1-q)} \right). \]
Proof continued

Other direction...

1. \( \mu(k) = \binom{n}{k} q^k (1 - q)^{n-k} \)
2. \( \sum_{i=0}^{n} \binom{n}{i} q^i (1 - q)^{n-i} = \sum_{i=0}^{n} \mu(i) \).
3. Claim: \( \mu(nq) = \binom{n}{nq} q^{nq}(1 - q)^{n-nq} \) largest term in \( \sum_{k=0}^{n} \mu(k) = 1 \).
4. \( \Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1 - q)^{n-k} \left( 1 - \frac{n-k}{k+1} \frac{q}{1-q} \right) \),
5. sign of \( \Delta_k \) = size of last term...
6. sign(\( \Delta_k \)) = sign\( \left(1 - \frac{(n-k)q}{(k+1)(1-q)}\right)\) 
   \[= \text{sign}\left(\frac{(k+1)(1-q) - (n-k)q}{(k+1)(1-q)}\right)\].
Proof continued

Other direction...

1. \( \mu(k) = \binom{n}{k} q^k (1 - q)^{n-k} \)
2. \( \sum_{i=0}^{n} \binom{n}{i} q^i (1 - q)^{n-i} = \sum_{i=0}^{n} \mu(i) \).
3. Claim: \( \mu(nq) = \binom{n}{nq} q^{nq} (1 - q)^{n-nq} \) largest term in \( \sum_{k=0}^{n} \mu(k) = 1 \).
4. \( \Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1 - q)^{n-k} \left(1 - \frac{n-k}{k+1} \frac{q}{1-q}\right) \),
5. sign of \( \Delta_k \) = size of last term...
6. \( \text{sign}(\Delta_k) = \text{sign}(1 - \frac{(n-k)q}{(k+1)(1-q)}) \)
   \( = \text{sign}\left(\frac{(k+1)(1-q)-(n-k)q}{(k+1)(1-q)}\right) \).
Proof continued

Other direction...

1. \( \mu(k) = \binom{n}{k} q^k (1 - q)^{n-k} \)
2. \( \sum_{i=0}^{n} \binom{n}{i} q^i (1 - q)^{n-i} = \sum_{i=0}^{n} \mu(i) \).
3. Claim: \( \mu(nq) = \binom{n}{nq} q^{nq} (1 - q)^{n-nq} \) largest term in \( \sum_{k=0}^{n} \mu(k) = 1 \).
4. \( \Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1 - q)^{n-k} \left( 1 - \frac{n-k}{k+1} \frac{q}{1-q} \right) \),
5. sign of \( \Delta_k \) = size of last term...
6. \( \text{sign}(\Delta_k) = \text{sign} \left( 1 - \frac{(n-k)q}{(k+1)(1-q)} \right) = \text{sign} \left( \frac{(k+1)(1-q)-(n-k)q}{(k+1)(1-q)} \right) \).
Proof continued

Other direction...

1. \( \mu(k) = \binom{n}{k} q^k (1 - q)^{n-k} \)

2. \( \sum_{i=0}^{n} \binom{n}{i} q^i (1 - q)^{n-i} = \sum_{i=0}^{n} \mu(i) \).

3. Claim: \( \mu(nq) = \binom{n}{nq} q^{nq} (1 - q)^{n-nq} \) largest term in \( \sum_{k=0}^{n} \mu(k) = 1 \).

4. \( \Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1 - q)^{n-k} \left( 1 - \frac{n-k}{k+1} \frac{q}{1-q} \right) \),

5. sign of \( \Delta_k \) = size of last term...

6. \( \text{sign}(\Delta_k) = \text{sign} \left( 1 - \frac{(n-k)q}{(k+1)(1-q)} \right) = \text{sign} \left( \frac{(k+1)(1-q) - (n-k)q}{(k+1)(1-q)} \right) \).
Proof continued

Other direction...

1. \( \mu(k) = \binom{n}{k} q^k (1 - q)^{n-k} \)

2. \[ \sum_{i=0}^{n} \binom{n}{i} q^i (1 - q)^{n-i} = \sum_{i=0}^{n} \mu(i). \]

3. Claim: \( \mu(nq) = \binom{n}{nq} q^{nq} (1 - q)^{n-nq} \) largest term in \( \sum_{k=0}^{n} \mu(k) = 1. \)

4. \( \Delta_k = \mu(k) - \mu(k + 1) = \binom{n}{k} q^k (1 - q)^{n-k} \left( 1 - \frac{n-k}{k+1} q \frac{1}{1-q} \right), \)

5. sign of \( \Delta_k = \) size of last term...

6. \( \text{sign}(\Delta_k) = \text{sign} \left( 1 - \frac{(n-k)q}{(k+1)(1-q)} \right) \)

\[ = \text{sign} \left( \frac{(k+1)(1-q)-(n-k)q}{(k+1)(1-q)} \right). \]
Proof continued

1. \[(k + 1)(1 - q) - (n - k)q = \frac{k + 1 - kq - q - nq + kq}{k + 1 - q} = 1 + k - q - nq.\]

2. \[\Rightarrow \Delta_k \geq 0 \text{ when } k \geq nq + q - 1 \]
\[\Delta_k < 0 \text{ otherwise.}\]

3. \[\mu(k) = \binom{n}{k} q^k (1 - q)^{n-k}\]

4. \[\mu(k) < \mu(k + 1), \text{ for } k < nq, \text{ and} \]
\[\mu(k) \geq \mu(k + 1) \text{ for } k \geq nq.\]

5. \[\Rightarrow \mu(nq) \text{ is the largest term in} \]
\[\sum_{k=0}^{n} \mu(k) = 1.\]

6. \[\mu(nq) \text{ larger than the average in sum.}\]

7. \[\Rightarrow \binom{n}{k} q^k (1 - q)^{n-k} \geq \frac{1}{n+1}.\]

8. \[\Rightarrow \]
Proof continued

1. \((k + 1)(1 - q) - (n - k)q = (k + 1 - kq - q - nq + kq) = 1 + k - q - nq.\)

2. \(\iff \Delta_k \geq 0 \text{ when } k \geq nq + q - 1 \quad \Delta_k < 0 \text{ otherwise.}\)

3. \(\mu(k) = \binom{n}{k}q^k(1 - q)^{n-k}\)

4. \(\mu(k) < \mu(k + 1), \text{ for } k < nq, \text{ and } \mu(k) \geq \mu(k + 1) \text{ for } k \geq nq.\)

5. \(\iff \mu(nq) \text{ is the largest term in } \sum_{k=0}^{n} \mu(k) = 1.\)

6. \(\mu(nq) \text{ larger than the average in sum.}\)

7. \(\iff \binom{n}{k}q^k(1 - q)^{n-k} \geq \frac{1}{n+1}.\)

8. \(\iff\)
Proof continued

1. \((k + 1)(1 - q) - (n - k)q = \)
   \[ k + 1 - kq - q - nq + kq = 1 + k - q - nq. \]
2. \(\implies \Delta_k \geq 0 \text{ when } k \geq nq + q - 1 \)
   \(\Delta_k < 0 \text{ otherwise.} \)
3. \(\mu(k) = \binom{n}{k}q^k(1 - q)^{n-k} \)
4. \(\mu(k) < \mu(k + 1), \text{ for } k < nq, \text{ and} \)
   \(\mu(k) \geq \mu(k + 1) \text{ for } k \geq nq. \)
5. \(\implies \mu(nq) \text{ is the largest term in} \)
   \[ \sum_{k=0}^{n} \mu(k) = 1. \]
6. \(\mu(nq) \text{ larger than the average in sum.} \)
7. \(\implies \binom{n}{k}q^k(1 - q)^{n-k} \geq \frac{1}{n+1}. \)
8. \(\implies \)
1. \[(k + 1)(1 - q) - (n - k)q = k + 1 - kq - q - nq + kq = 1 + k - q - nq.\]

2. \[\iff \Delta_k \geq 0 \text{ when } k \geq nq + q - 1\]
   \[\Delta_k < 0 \text{ otherwise.}\]

3. \[\mu(k) = \binom{n}{k} q^k (1 - q)^{n-k}\]

4. \[\mu(k) < \mu(k + 1), \text{ for } k < nq, \text{ and}\]
   \[\mu(k) \geq \mu(k + 1) \text{ for } k \geq nq.\]

5. \[\iff \mu(nq) \text{ is the largest term in } \sum_{k=0}^{n} \mu(k) = 1.\]

6. \[\mu(nq) \text{ larger than the average in sum.}\]

7. \[\iff \binom{n}{k} q^k (1 - q)^{n-k} \geq \frac{1}{n+1}.\]
Proof continued

1. \((k + 1)(1 - q) - (n - k)q = k + 1 - kq - q - nq + kq = 1 + k - q - nq\).

2. \(\Rightarrow \Delta_k \geq 0\) when \(k \geq nq + q - 1\)
   \(\Delta_k < 0\) otherwise.

3. \(\mu(k) = \binom{n}{k}q^k(1 - q)^{n-k}\)

4. \(\mu(k) < \mu(k + 1)\), for \(k < nq\), and
   \(\mu(k) \geq \mu(k + 1)\) for \(k \geq nq\).

5. \(\Rightarrow \mu(nq)\) is the largest term in
   \(\sum_{k=0}^{n} \mu(k) = 1\).

6. \(\mu(nq)\) larger than the average in sum.

7. \(\Rightarrow \binom{n}{k}q^k(1 - q)^{n-k} \geq \frac{1}{n+1}\).

8. \(\Rightarrow \frac{n}{k} \frac{q^{k-1}}{1 - q} \frac{1}{n-k+1} \geq \frac{1}{n+1}\).
Proof continued

1. \((k + 1)(1 - q) - (n - k)q = k + 1 - kq - q - nq + kq = 1 + k - q - nq.\)
2. \(\implies \Delta_k \geq 0 \text{ when } k \geq nq + q - 1\)
\(\Delta_k < 0 \text{ otherwise.}\)
3. \(\mu(k) = \binom{n}{k} q^k (1 - q)^{n-k}\)
4. \(\mu(k) < \mu(k + 1), \text{ for } k < nq, \text{ and}\)
\(\mu(k) \geq \mu(k + 1) \text{ for } k \geq nq.\)
5. \(\implies \mu(nq) \text{ is the largest term in } \sum_{k=0}^{n} \mu(k) = 1.\)
6. \(\mu(nq) \text{ larger than the average in sum.}\)
7. \(\implies \binom{n}{k} q^k (1 - q)^{n-k} \geq \frac{1}{n+1}.\)
Proof continued

1. \((k + 1)(1 - q) - (n - k)q = k + 1 - kq - q - nq + kq = 1 + k - q - nq.\)

2. \(\implies \Delta_k \geq 0\) when \(k \geq nq + q - 1\)
   \(\Delta_k < 0\) otherwise.

3. \(\mu(k) = \binom{n}{k}q^k(1 - q)^{n-k}\)

4. \(\mu(k) < \mu(k + 1)\), for \(k < nq\), and
   \(\mu(k) \geq \mu(k + 1)\) for \(k \geq nq\).

5. \(\implies \mu(nq)\) is the largest term in
   \(\sum_{k=0}^{n} \mu(k) = 1.\)

6. \(\mu(nq)\) larger than the average in sum.

7. \(\implies \binom{n}{k}q^k(1 - q)^{n-k} \geq \frac{1}{n+1}.\)
Proof continued

1. \((k + 1)(1 - q) - (n - k)q = k + 1 - kq - q - nq + kq = 1 + k - q - nq.\)

2. \(\implies \Delta_k \geq 0 \text{ when } k \geq nq + q - 1\)
   \(\Delta_k < 0 \text{ otherwise.}\)

3. \(\mu(k) = \binom{n}{k}q^k(1 - q)^{n-k}\)

4. \(\mu(k) < \mu(k + 1), \text{ for } k < nq, \text{ and }\)
   \(\mu(k) \geq \mu(k + 1) \text{ for } k \geq nq.\)

5. \(\implies \mu(nq) \text{ is the largest term in }\)
   \(\sum_{k=0}^{n} \mu(k) = 1.\)

6. \(\mu(nq) \text{ larger than the average in sum.}\)

7. \(\implies \binom{n}{k}q^k(1 - q)^{n-k} \geq \frac{1}{n+1}.\)

8. \(\implies\)
Flipper revisited...

Clicker question

\( p \): coin returns head with this probability. \( q = 1 - p \).

Flip coin \( n \) times, let \( X \) be the resulting string. Assume \( np \) and \( nq \) are integer.

\( S_i \): set of all binary strings length \( n \) with \( i \) ones in them. Then:

1. \( \Pr[X \in S_i] \) is maximal for \( i = np \).
2. \( \forall s, s' \in S_i \), we have
   \[
   \Pr[X = s] = \Pr[X = s'] = \binom{n}{i} p^i q^{n-i}.
   \]
3. If \( X \in S_i \) then entropy of \( X \) is \( \lg \binom{n}{i} \).
4. \( H(X) = n \mathbb{H}(p) \)
5. All of the above.
Generalization...

Corollary

We have:

1. $q \in [0, 1/2] \Rightarrow \left( \left\lfloor \frac{n}{nq} \right\rfloor \right) \leq 2^{n \mathbb{H}(q)}$.
2. $q \in [1/2, 1] \Rightarrow \left( \left\lceil \frac{n}{nq} \right\rceil \right) \leq 2^{n \mathbb{H}(q)}$.
3. $q \in [1/2, 1] \Rightarrow \frac{2^{n \mathbb{H}(q)} \cdot n}{n+1} \leq \binom{n}{\left\lfloor nq \right\rfloor}$.
4. $q \in [0, 1/2] \Rightarrow \frac{2^{n \mathbb{H}(q)} \cdot n}{n+1} \leq \binom{n}{\left\lceil nq \right\rceil}$.

Proof is straightforward but tedious.
What we have...

1. Proved that \( \binom{n}{nq} \approx 2^n \mathbb{H}(q) \).
2. Estimate is loose.
3. Sanity check...
   
   3.1 A sequence of \( n \) bits generated by coin with probability \( q \) for head.
   
   3.2 By Chernoff inequality... roughly \( nq \) heads in this sequence.
   
   3.3 Generated sequence \( Y \) belongs to \( \binom{n}{nq} \approx 2^n \mathbb{H}(q) \) possible sequences.
   
   3.4 ...of similar probability.
   
   3.5 \( \implies \mathbb{H}(Y) = n \mathbb{H}(q) \approx \log \binom{n}{nq} \).
What we have...

1. Proved that $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$.
2. Estimate is loose.
3. Sanity check...

3.1 A sequence of $n$ bits generated by coin with probability $q$ for head.
3.2 By Chernoff inequality... roughly $nq$ heads in this sequence.
3.3 Generated sequence $Y$ belongs to $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$ possible sequences.
3.4 ...of similar probability.
3.5 $\implies \mathbb{H}(Y) = n\mathbb{H}(q) \approx \lg \binom{n}{nq}$. 
What we have...

1. Proved that \( \binom{n}{nq} \approx 2^{n\mathbb{H}(q)} \).
2. Estimate is loose.
3. Sanity check...
   3.1 A sequence of \( n \) bits generated by coin with probability \( q \) for head.
   3.2 By Chernoff inequality... roughly \( nq \) heads in this sequence.
   3.3 Generated sequence \( Y \) belongs to \( \binom{n}{nq} \approx 2^{n\mathbb{H}(q)} \) possible sequences .
   3.4 ...of similar probability.
   3.5 \( \mathbb{H}(Y) = n\mathbb{H}(q) \approx \log \binom{n}{nq} \).
What we have...

1. Proved that $\binom{n}{nq} \approx 2^{n \mathbb{H}(q)}$.
2. Estimate is loose.
3. Sanity check...
   3.1 A sequence of $n$ bits generated by coin with probability $q$ for head.
   3.2 By Chernoff inequality... roughly $nq$ heads in this sequence.
   3.3 Generated sequence $Y$ belongs to $\binom{n}{nq} \approx 2^{n \mathbb{H}(q)}$ possible sequences.
   3.4 ...of similar probability.
   3.5 $\Rightarrow \mathbb{H}(Y) = n \mathbb{H}(q) \approx \lg \binom{n}{nq}$. 
What we have...

1. Proved that \( \binom{n}{nq} \approx 2^{nH(q)} \).
2. Estimate is loose.
3. Sanity check...
   3.1 A sequence of \( n \) bits generated by coin with probability \( q \) for head.
   3.2 By Chernoff inequality... roughly \( nq \) heads in this sequence.
   3.3 Generated sequence \( Y \) belongs to \( \binom{n}{nq} \approx 2^{nH(q)} \) possible sequences .
   3.4 ...of similar probability.
   3.5 \( \implies H(Y) = nH(q) \approx \lg \binom{n}{nq} \).
Just one bit...

question
Given a coin $C$ with:
$p$: Probability for head.
$q = 1 - p$: Probability for tail.
$Q$: How to get one true random bit, by flipping $C$.
Describe an algorithm!
Extracting randomness...

Entropy can be interpreted as the amount of unbiased random coin flips can be extracted from a random variable.

Definition

An extraction function $\text{Ext}$ takes as input the value of a random variable $X$ and outputs a sequence of bits $y$, such that $\Pr[\text{Ext}(X) = y \mid |y| = k] = \frac{1}{2^k}$, whenever $\Pr[|y| = k] > 0$, where $|y|$ denotes the length of $y$. 
Extracting randomness...

1. \( X \): uniform random integer variable out of \( 0, \ldots, 7 \).
2. \( \text{Ext}(X) \): binary representation of \( x \).
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
   4.1 \( X \): uniform random integer variable \( 0, \ldots, 11 \).
   4.2 \( \text{Ext}(x) \): output the binary representation for \( x \) if \( 0 \leq x \leq 7 \).
   4.3 If \( x \) is between 8 and 11?
   4.4 Idea... Output binary representation of \( x - 8 \) as a two bit number.
5. A valid extractor...
Extracting randomness...

1. **$X$**: uniform random integer variable out of $0, \ldots, 7$.
2. **$\text{Ext}(X)$**: binary representation of $x$.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
   4.1 **$X$**: uniform random integer variable $0, \ldots, 11$.
   4.2 **$\text{Ext}(x)$**: output the binary representation for $x$ if $0 \leq x \leq 7$.
   4.3 If $x$ is between 8 and 11?
   4.4 Idea... Output binary representation of $x - 8$ as a two bit number.
5. A valid extractor...

\[
\Pr[\text{Ext}(X) = 00 | \text{Ext}(X) = 2] = \frac{32}{35}.
\]
Extracting randomness...

1. \( X \): uniform random integer variable out of \( 0, \ldots, 7 \).
2. \( \text{Ext}(X) \): binary representation of \( x \).
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
   4.1 \( X \): uniform random integer variable \( 0, \ldots, 11 \).
   4.2 \( \text{Ext}(x) \): output the binary representation for \( x \) if \( 0 \leq x \leq 7 \).
   4.3 If \( x \) is between 8 and 11?
   4.4 Idea... Output binary representation of \( x - 8 \) as a two bit number.
5. A valid extractor...
Extracting randomness...

1. $X$: uniform random integer variable out of $0, \ldots, 7$.
2. $\text{Ext}(X)$: binary representation of $x$.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
   4.1 $X$: uniform random integer variable $0, \ldots, 11$.
   4.2 $\text{Ext}(x)$: output the binary representation for $x$ if $0 \leq x \leq 7$.
   4.3 If $x$ is between $8$ and $11$?
   4.4 Idea... Output binary representation of $x - 8$ as a two bit number.
5. A valid extractor...
   \[ \Pr[\text{Ext}(X) = 00 \mid \text{Ext}(X) = 2] = \frac{32}{35} \]
Extracting randomness...

1. $X$: uniform random integer variable out of $0, \ldots, 7$.
2. $\text{Ext}(X)$: binary representation of $x$.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
   4.1 $X$: uniform random integer variable $0, \ldots, 11$.
   4.2 $\text{Ext}(x)$: output the binary representation for $x$ if $0 \leq x \leq 7$.
   4.3 If $x$ is between 8 and 11?
   4.4 Idea... Output binary representation of $x - 8$ as a two bit number.
5. A valid extractor...
Extracting randomness...

1. \( X \): uniform random integer variable out of \( 0, \ldots, 7 \).
2. \( \text{Ext}(X) \): binary representation of \( x \).
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
   4.1 \( X \): uniform random integer variable \( 0, \ldots, 11 \).
   4.2 \( \text{Ext}(x) \): output the binary representation for \( x \) if \( 0 \leq x \leq 7 \).
   4.3 If \( x \) is between \( 8 \) and \( 11 \)?
   4.4 Idea... Output binary representation of \( x - 8 \) as a two bit number.
5. A valid extractor...
Extracting randomness...

1. **X**: uniform random integer variable out of 0, . . . , 7.
2. **Ext(X)**: binary representation of **x**.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
   4.1 **X**: uniform random integer variable 0, . . . , 11.
   4.2 **Ext(x)**: output the binary representation for **x** if 0 ≤ **x** ≤ 7.
   4.3 If **x** is between 8 and 11?
   4.4 Idea... Output binary representation of **x** − 8 as a two bit number.

5. A valid extractor...
Extracting randomness...

1. $X$: uniform random integer variable out of $0, \ldots, 7$.
2. $\text{Ext}(X)$: binary representation of $x$.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
   4.1 $X$: uniform random integer variable $0, \ldots, 11$.
   4.2 $\text{Ext}(x)$: output the binary representation for $x$ if $0 \leq x \leq 7$.
   4.3 If $x$ is between 8 and 11?
   4.4 Idea... Output binary representation of $x - 8$ as a two bit number.
5. A valid extractor...
Technical lemma

The following is obvious, but we provide a proof anyway.

Lemma

Let $x/y$ be a faction, such that $x/y < 1$. Then, for any $i$, we have $x/y < (x + i)/(y + i)$.

Proof.

We need to prove that $x(y + i) - (x + i)y < 0$. The left size is equal to $i(x - y)$, but since $y > x$ (as $x/y < 1$), this quantity is negative, as required. $\square$
A uniform variable extractor...

Theorem

1. \( X \): random variable chosen uniformly at random from \( \{0, \ldots, m - 1\} \).
2. Then there is an extraction function for \( X \):
   2.1 outputs on average at least
   \[\lfloor \log m \rfloor - 1 = \lfloor H(X) \rfloor - 1\]
   independent and unbiased bits.
A uniform variable extractor...

Theorem

1. \( X \): random variable chosen uniformly at random from \( \{0, \ldots, m - 1\} \).

2. Then there is an extraction function for \( X \):

   2.1 outputs on average at least

   \[
   \lfloor \lg m \rfloor - 1 = \lfloor H(X) \rfloor - 1
   \]

   independent and unbiased bits.
A uniform variable extractor...

**Theorem**

1. **$X$**: random variable chosen uniformly at random from $\{0, \ldots, m - 1\}$.

2. Then there is an extraction function for $X$:
   2.1 outputs on average at least
   
   $$\lceil \log m \rceil - 1 = \lfloor H(X) \rfloor - 1$$

   independent and unbiased bits.
Proof

1. \( m \): A sum of unique powers of 2, namely 
   \( m = \sum_i a_i 2^i \), where \( a_i \in \{0, 1\} \).

2. Example:

3. decomposed \( \{0, \ldots, m - 1\} \) into disjoint union of blocks sizes are powers of 2.

4. If \( x \) is in block \( 2^k \), output its relative location in the block in binary representation.

5. Example: \( x = 10 \):
   then falls into block \( 2^2 \)... 
   \( x \) relative location is 2. Output 2 written using two bits,
   Output: “10”. 
Proof

1. \( m \): A sum of unique powers of 2, namely
   \[ m = \sum_i a_i 2^i, \text{ where } a_i \in \{0, 1\}. \]

2. Example:

3. decomposed \( \{0, \ldots, m - 1\} \) into disjoint union of blocks sizes are powers of 2.

4. If \( x \) is in block \( 2^k \), output its relative location in the block in binary representation.

5. Example: \( x = 10 \):
   then falls into block \( 2^2 \)...
   \( x \) relative location is 2. Output 2 written using two bits.
Proof

1. \( m \): A sum of unique powers of 2, namely 
   \[ m = \sum_i a_i 2^i, \text{ where } a_i \in \{0, 1\}. \]

2. Example:

3. decomposed \( \{0, \ldots, m - 1\} \) into disjoint union of blocks sizes are powers of 2.

4. If \( x \) is in block \( 2^k \), output its relative location in the block in binary representation.

5. Example: \( x = 10 \):
   then falls into block \( 2^2 \)...
   relative location in 2: Output 2 written in 5.
Proof

1. $m$: A sum of unique powers of $2$, namely $m = \sum_i a_i 2^i$, where $a_i \in \{0, 1\}$.

2. Example:

3. decomposed $\{0, \ldots, m - 1\}$ into disjoint union of blocks sizes are powers of $2$.

4. If $x$ is in block $2^k$, output its relative location in the block in binary representation.

5. Example: $x = 10$: then falls into block $2^2$... relative location in 2. Output: $010$ written in 3.
Proof

1. \( m \): A sum of unique powers of 2, namely 
   \[ m = \sum_i a_i 2^i, \text{ where } a_i \in \{0, 1\}. \]

2. Example:

3. decomposed \( \{0, \ldots, m - 1\} \) into disjoint union of blocks sizes are powers of 2.

4. If \( x \) is in block \( 2^k \), output its relative location in the block in binary representation.

5. Example: \( x = 10 \):
Proof

1. \( m \): A sum of unique powers of 2, namely \( m = \sum_i a_i 2^i \), where \( a_i \in \{0, 1\} \).

2. Example:

3. decomposed \( \{0, \ldots, m - 1\} \) into disjoint union of blocks sizes are powers of 2.

4. If \( x \) is in block \( 2^k \), output its relative location in the block in binary representation.

5. Example: \( x = 10 \):
Proof

1. \( m \): A sum of unique powers of 2, namely 
\[ m = \sum_i a_i 2^i, \text{ where } a_i \in \{0, 1\}. \]

2. Example:

\[
\begin{array}{cccccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\
\end{array}
\]

3. decomposed \( \{0, \ldots, m - 1\} \) into disjoint union of blocks sizes are powers of 2.

4. If \( x \) is in block \( 2^k \), output its relative location in the block in binary representation.

5. Example: \( x = 10 \):
Proof

1. \( m \): A sum of unique powers of 2, namely
\[
m = \sum_{i} a_i 2^i,
\]
where \( a_i \in \{0, 1\} \).

2. Example:

\[
\begin{array}{cccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
11 & 12 & 13
\end{array}
\]

3. decomposed \( \{0, \ldots, m - 1\} \) into disjoint union of blocks sizes are powers of 2.

4. If \( x \) is in block \( 2^k \), output its relative location in the block in binary representation.

5. Example: \( x = 10 \):

\[
\begin{array}{cccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
11 & 12 & 13
\end{array}
\]

Output: "10".
Proof continued

1. Valid extractor...
2. Theorem holds if $m$ is a power of two. Only one block.
3. $m$ not a power of $2$...
4. $X$ falls in block of size $2^k$: then output $k$ complete random bits...
   ... entropy is $k$.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \log(m - 2^k) \rfloor < k$.
   There must be a block of size $u$ in the decomposition of $m$.
7. two blocks in decomposition of $m$: sizes $2^k$ and $2^u$.
8. Largest two blocks...
9. $2^k + 2 + 2^u > \ldots \geq 2^{u+1} + 2^k \implies u > 0$. 

Proof continued

1. Valid extractor...
2. Theorem holds if $m$ is a power of two. Only one block.
3. $m$ not a power of 2...
4. $X$ falls in block of size $2^k$: then output $k$ complete random bits.
   ... entropy is $k$.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.
   There must be a block of size $u$ in the decomposition of $m$.
7. two blocks in decomposition of $m$: sizes $2^k$ and $2^u$.
8. Largest two blocks...
9. $2^k + 2 + 2^u > \ldots > 2^{u+1} + 2^k \ldots m > 0$.
Proof continued

1. Valid extractor...
2. Theorem holds if $m$ is a power of two. Only one block.
3. $m$ not a power of 2...
4. $X$ falls in block of size $2^k$: then output $k$ complete random bits...
   ... entropy is $k$.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.
   There must be a block of size $u$ in the decomposition of $m$.
7. two blocks in decomposition of $m$: sizes $2^k$ and $2^u$.
8. Largest two blocks...
9. $2^k + 2 + 2^u > m \implies 2^{u+1} + 2^k - m > 0$. 

Proof continued

1. Valid extractor...
2. Theorem holds if \( m \) is a power of two. Only one block.
3. \( m \) not a power of \( 2 \)...
4. \( X \) falls in block of size \( 2^k \): then output \( k \) complete random bits...
   ... entropy is \( k \).
5. Let \( 2^k < m < 2^{k+1} \) biggest block.
6. \( u = \lceil \lg(m - 2^k) \rceil < k \).
   There must be a block of size \( u \) in the decomposition of \( m \).
7. two blocks in decomposition of \( m \): sizes \( 2^k \) and \( 2^u \).
8. Largest two blocks...
9. \( 2^k + 2 + 2^u > \ldots > 2^{u+1} + 2^k \Rightarrow u > 0 \).
Proof continued

1. Valid extractor...
2. Theorem holds if $m$ is a power of two. Only one block.
3. $m$ not a power of 2...
4. $X$ falls in block of size $2^k$: then output $k$ complete random bits..
   ... entropy is $k$.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.
   There must be a block of size $u$ in the decomposition of $m$.
7. two blocks in decomposition of $m$: sizes $2^k$ and $2^u$.
8. Largest two blocks...
9. $2^k + 2^u + 2^{u+1} + \ldots + 2^{u+1} + 2^{k-u} > m > 0$. 
Proof continued

1. Valid extractor...
2. Theorem holds if $m$ is a power of two. Only one block.
3. $m$ not a power of $2$...
4. $X$ falls in block of size $2^k$: then output $k$ complete random bits. 
   ... entropy is $k$.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \log (m - 2^k) \rfloor < k$. 
   There must be a block of size $u$ in the decomposition of $m$.
7. two blocks in decomposition of $m$: sizes $2^k$ and $2^u$.
8. Largest two blocks...
9. $2^k \cdot 2 + 2^u > m \Rightarrow 2^u + 1 + 2^k \Rightarrow m > 0$. 
Proof continued

1. Valid extractor...
2. Theorem holds if $m$ is a power of two. Only one block.
3. $m$ not a power of $2$...
4. $X$ falls in block of size $2^k$: then output $k$ complete random bits.
   ... entropy is $k$.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \log_2(m - 2^k) \rfloor < k$.
   There must be a block of size $u$ in the decomposition of $m$.
7. Two blocks in decomposition of $m$: sizes $2^k$ and $2^u$.
8. Largest two blocks...
9. $2^k + 2 + 2^u > \cdots > 2^{u+1} + 2^k \Rightarrow m > 0$.
Proof continued

1. Valid extractor...
2. Theorem holds if $m$ is a power of two. Only one block.
3. $m$ not a power of 2...
4. $X$ falls in block of size $2^k$: then output $k$ complete random bits...
   ... entropy is $k$.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \log(m - 2^k) \rfloor < k$.
   There must be a block of size $u$ in the decomposition of $m$.
7. two blocks in decomposition of $m$: sizes $2^k$ and $2^u$.
8. Largest two blocks...
Proof continued

1. Valid extractor...
2. Theorem holds if $m$ is a power of two. Only one block.
3. $m$ not a power of $2$...
4. $X$ falls in block of size $2^k$: then output $k$ complete random bits...
   ... entropy is $k$.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \log(m - 2^k) \rfloor < k$.
   There must be a block of size $u$ in the decomposition of $m$.
7. two blocks in decomposition of $m$: sizes $2^k$ and $2^u$.
8. Largest two blocks...
Proof continued

1. Valid extractor...
2. Theorem holds if $m$ is a power of two. Only one block.
3. $m$ not a power of $2$...
4. $X$ falls in block of size $2^k$: then output $k$ complete random bits..
   ... entropy is $k$.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.
   There must be a block of size $u$ in the decomposition of $m$.
7. two blocks in decomposition of $m$: sizes $2^k$ and $2^u$.
8. Largest two blocks...
9. $2^k + 2 + 2^u > m \Rightarrow 2^u + 1 + 2^k \Rightarrow m > 0.$
Proof continued

1. Valid extractor...
2. Theorem holds if $m$ is a power of two. Only one block.
3. $m$ not a power of $2$...
4. $X$ falls in block of size $2^k$: then output $k$ complete random bits..
   ... entropy is $k$.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lceil \log(m - 2^k) \rceil < k$.
   There must be a block of size $u$ in the decomposition of $m$.
7. two blocks in decomposition of $m$: sizes $2^k$ and $2^u$.
8. Largest two blocks...
9. $2^k + 2 + 2^u > m \Rightarrow 2^{u+1} + 2^k - m > 0$. 


Proof continued

1. By lemma, since $\frac{m-2^k}{m} < 1$:

$$\frac{m - 2^k}{m} \leq \frac{m - 2^k + (2^{u+1} + 2^k - m)}{m} + (2^{u+1} + 2^k - m) = \frac{2^{u+1}}{2^{u+1} + 2^k}.$$ 

2. By induction (assumed holds for all numbers smaller than $m$):

$$E[Y] \geq \frac{2^k}{m}k + \frac{m - 2^k}{m} \left( \left\lceil \lg(m - 2^k) \right\rceil - 1 \right)$$

$$= \frac{2^k}{m}k + \frac{m - 2^k}{m} \left( k - k + u - 1 \right) = 0.$$
Proof continued

1. By lemma, since \( \frac{m - 2^k}{m} < 1 \):

\[
\frac{m - 2^k}{m} \leq \frac{m - 2^k}{m} + \left( 2^u + 1 + 2^k - m \right) = \frac{2^u + 1}{2^u + 1 + 2^k}.
\]

2. By induction (assumed holds for all numbers smaller than \( m \)):

\[
\mathbb{E}[Y] \geq 2^k \cdot k + \frac{m - 2^k}{m} \left( \left\lfloor \log_2(m - 2^k) \right\rfloor - 1 \right) \]

\[
= \frac{2^k}{m} k + \frac{m - 2^k}{m} (k - k + u - 1) = 0
\]
Proof continued

1. By lemma, since \(\frac{m - 2^k}{m} < 1\):

\[
\frac{m - 2^k}{m} \leq \frac{m - 2^k + (2^{u+1} + 2^k - m)}{m} + (2^{u+1} + 2^k - m) = \frac{2^{u+1}}{2^{u+1} + 2^k}.
\]

2. By induction (assumed holds for all numbers smaller than \(m\)):

\[
E[Y] \geq \frac{2^k}{m} k + \frac{m - 2^k}{m} \left(\left\lfloor \log_2(m - 2^k) \right\rfloor - 1 \right)
\]

\[
= \frac{2^k}{m} k + \frac{m - 2^k}{m} (k - k + u - 1)
\]

\[
= 0
\]
Proof continued

1. By lemma, since \( \frac{m-2^k}{m} < 1 \):

\[
\frac{m - 2^k}{m} \leq \frac{m - 2^k + (2^{u+1} + 2^k - m)}{m} + (2^{u+1} + 2^k - m) = \frac{2^{u+1}}{2^{u+1} + 2^k}.
\]

2. By induction (assumed holds for all numbers smaller than \( m \)):

\[
\mathbb{E}[Y] \geq \frac{2^k}{m} k + \frac{m - 2^k}{m} \left( \left\lfloor \log(m - 2^k) \right\rfloor - 1 \right) = 0
\]
Proof continued..

1. We have:

\[
E[Y] \geq k + \frac{m - 2^k}{m} (u - k - 1)
\]

\[
\geq k + \frac{2^{u+1}}{2^{u+1} + 2^k} (u - k - 1)
\]

\[
= k - \frac{2^{u+1}}{2^{u+1} + 2^k} (1 + k - u),
\]

since \( u - k - 1 \leq 0 \) as \( k > u \).

2. If \( u = k - 1 \), then \( E[Y] \geq k - \frac{1}{2} \cdot 2 = k - 1 \), as required.

3. If \( u = k - 2 \), then \( E[Y] \geq k - \frac{1}{3} \cdot 3 = k - 1 \).
Proof continued..

1. We have:

\[
E[Y] \geq k + \frac{m - 2^k}{m} (u - k - 1)
\geq k + \frac{2^{u+1}}{2^{u+1} + 2^k} (u - k - 1)
= k - \frac{2^{u+1}}{2^{u+1} + 2^k} (1 + k - u),
\]

since \( u - k - 1 \leq 0 \) as \( k > u \).

2. If \( u = k - 1 \), then \( E[Y] \geq k - \frac{1}{2} \cdot 2 = k - 1 \), as required.

3. If \( u = k - 2 \) then \( E[Y] \geq k - \frac{1}{3} \cdot 3 = k - 1 \).
Proof continued..

1. We have:

\[
E[Y] \geq k + \frac{m - 2^k}{m}(u - k - 1)
\]

\[
\geq k + \frac{2^{u+1}}{2^{u+1} + 2^k}(u - k - 1)
\]

\[
= k - \frac{2^{u+1}}{2^{u+1} + 2^k}(1 + k - u),
\]

since \(u - k - 1 \leq 0\) as \(k > u\).

2. If \(u = k - 1\), then \(E[Y] \geq k - \frac{1}{2} \cdot 2 = k - 1\), as required.

3. If \(u = k - 2\) then \(E[Y] \geq k - \frac{1}{3} \cdot 3 = k - 1\).
Proof continued..

1. We have:

\[ \mathbb{E}[Y] \geq k + \frac{m - 2^k}{m} (u - k - 1) \]

\[ \geq k + \frac{2^{u+1}}{2^{u+1} + 2^k} (u - k - 1) \]

\[ = k - \frac{2^{u+1}}{2^{u+1} + 2^k} (1 + k - u), \]

since \( u - k - 1 \leq 0 \) as \( k > u \).

2. If \( u = k - 1 \), then \( \mathbb{E}[Y] \geq k - \frac{1}{2} \cdot 2 = k - 1 \), as required.

3. If \( u = k - 2 \) then \( \mathbb{E}[Y] \geq k - \frac{1}{3} \cdot 3 = k - 1 \).
Proof continued.....

1. \( E[Y] \geq k - \frac{2^{u+1}}{2^{u+1} + 2^k} (1 + k - u) \).
   And \( u - k - 1 \leq 0 \) as \( k > u \).

2. If \( u < k - 2 \) then

\[
E[Y] \geq k - \frac{2^{u+1}}{2^k} (1 + k - u)
\]

\[
= k - \frac{k - u + 1}{2^{k-u-1}}
\]

\[
= k - \frac{2 + (k - u - 1)}{2^{k-u-1}}
\]

\[
\geq k - 1,
\]

since \((2 + i)/2^i \leq 1\) for \(i \geq 2\).
Proof continued.....

1. \( \mathbb{E}[Y] \geq k - \frac{2^{u+1}}{2u+1+2^k}(1 + k - u). \)
   And \( u - k - 1 \leq 0 \) as \( k > u. \)

2. If \( u < k - 2 \) then

   \[
   \mathbb{E}[Y] \geq k - \frac{2^{u+1}}{2^k}(1 + k - u) \\
   = k - \frac{k - u + 1}{2^{k-u-1}} \\
   = k - \frac{2 + (k - u - 1)}{2^{k-u-1}} \\
   \geq k - 1,
   \]

   since \((2 + i)/2^i \leq 1\) for \( i \geq 2. \)