Entropy and Shannon’s Theorem

Lecture 24
November 18, 2018
Part I

Entropy
Part II

Extracting randomness
How many binary strings are there with \( n \) bits and \( j \) ones.

(A) \( \binom{n-1}{j} \).

(B) \( \binom{n}{j-1} \).

(C) \( \binom{n}{j} \).

(D) \( 2^n \).

(E) \( 2^{\binom{n}{j}} \).
How many strings?
Clicker question

How many binary strings are there with $n$ bits and $j$ ones, where the first bit is zero.

(A) $\binom{n-1}{j}$.
(B) $\binom{n}{j-1}$.
(C) $\binom{n}{j}$.
(D) $2^n$.
(E) $2^{\binom{n}{j}}$. 
How many strings?

Clicker question

Assuming you have regular computer, and multiplying two 32 bit numbers takes constant time, and you can directly multiply larger numbers (i.e., you need to implement such operations yourself). Then, computing \( \binom{n}{j} \) (using a reasonably fast algorithm) takes time

(A) \( O\left(\binom{n}{j}\right) \).

(B) \( O\left((j \log n)^2\right) \).

(C) \( 2^n \).

(D) \( 2^{\binom{n}{j}} \).

(E) Polynomial time.
Storing all strings of length $n$ and $j$ bits on

1. $S_{n,j}$: set of all strings of length $n$ with $j$ ones in them.
2. $T_{n,j}$: prefix tree storing all $S_{n,j}$. 

![Diagram of prefix tree]
Binary strings of length 4

1.

Diagram representing binary strings of length 4.
Binary strings of length 4

1. \( S_{4,0} = \{0000\} \implies \#(0000) = 0. \)
Binary strings of length 4

1. \( S_{4,1} = \{0001, 0010, 0100, 1000\} \)
   \[\implies \#(0001) = 0.\]
   \[\#(0010) = 1.\]
   \[\#(0100) = 2.\]
   \[\#(1000) = 3.\]

2. 8/10
Binary strings of length 4

1. $S_{4,2} = \{0011, 0101, 0110, 1001, 1010, 1100\}$

   $\Rightarrow$

   - $\#(0011) = 0$
   - $\#(0101) = 1$
   - $\#(0110) = 2$
   - $\#(1001) = 3$
   - $\#(1010) = 4$
Binary strings of length 4

1. \( S_{4,3} = \{0111, 1011, 1101, 1110\} \)

\[\implies\]

\( \#(0111) = 0 \).

\( \#(1011) = 1 \).

\( \#(1101) = 2 \).

\( \#(1110) = 3 \).
Binary strings of length 4

1. $S_{4,4} = \{1111\}$

$$\implies \quad \#(1111) = 0.$$
Prefix tree $\forall$ binary strings of length $n$ with $j$ ones

$T_{n,j}$

$T_{n,0}$

$T_{n,n}$
Prefix tree ∀ binary strings of length $n$ with $j$ ones

$T_{n,j}$

# of leafs: $|T_{n,j}| = |T_{n-1,j}| + |T_{n-1,j-1}|$

$T_{n,0}$

$T_{n,n}$
Prefix tree $\forall$ binary strings of length $n$ with $j$ ones

$T_{n,j}$

# of leafs:

$|T_{n,j}| = |T_{n-1,j}| + |T_{n-1,j-1}|$

$n \choose j = (n-1) \choose j + (n-1) \choose j-1$
Prefix tree $\forall$ binary strings of length $n$ with $j$ ones

$T_{n,j}$

$\# $ of leafs:

$|T_{n,j}| = |T_{n-1,j}| + |T_{n-1,j-1}|$

$(\binom{n}{j}) = (\binom{n-1}{j}) + (\binom{n-1}{j-1})$

$\Rightarrow |T_{n,j}| = (\binom{n}{j})$.
Encoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
3. ≡ ordering leafs of $T_{n,j}$ from left to right.
4. Input: $s \in S_{n,j}$: compute index of $s$ in sorted set $S_{n,j}$.
5. $\text{EncodeBinomCoeff}(s)$ denote this polytime procedure.
Encoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering
3. $≡$ ordering leafs of $T_{n,j}$ from left to right.
4. Input: $s ∈ S_{n,j}$: compute index of $s$ in sorted set $S_{n,j}$.
5. $\text{EncodeBinomCoeff}(s)$ denote this polytime procedure.
Encoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering
3. $≡$ ordering leafs of $T_{n,j}$ from left to right.

$$T_{n,j}$$

4. Input: $s \in S_{n,j}$: compute index of $s$ in sorted set $S_{n,j}$. 
Encoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.

4. Input: $s \in S_{n,j}$: compute index of $s$ in sorted set $S_{n,j}$. 
Encoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.

4. Input: $s \in S_{n,j}$: compute index of $s$ in sorted set $S_{n,j}$.
Decoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.
4. $x \in \{1, \ldots, \binom{n}{j}\}$: compute $x$th string in $S_{n,j}$ in polytime.
5. $\text{DecodeBinomCoeff}(x)$ denote this procedure.
Decoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.
4. $x \in \{1, \ldots, \binom{n}{j}\}$: compute $x$th string in $S_{n,j}$ in polytime.
5. $\text{DecodeBinomCoeff}(x)$ denote this procedure.
Decoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.

$$T_{n,j}$$

4. $x \in \{1, \ldots, \binom{n}{j}\}$: compute $x$th string in $S_{n,j}$ in polytime.
Decoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering
3. ≡ ordering leafs of $T_{n,j}$ from left to right.

$$T_{n,j}$$

4. $x \in \{1, \ldots, \binom{n}{j}\}$: compute $x$th string in $S_{n,j}$ in polytime
Decoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.

4. $x \in \{1, \ldots, \binom{n}{j}\}$: compute $x$th string in $S_{n,j}$ in polytime.

Diagram:

```
   0
  / \
1   1

T_{n-1,j}  T_{n-1,j-1}
```
Encoding/decoding strings of $S_{n,j}$

Lemma

$S_{n,j}$: Set of binary strings of length $n$ with $j$ ones, sorted lexicographically.

1. **EncodeBinomCoeff**($\alpha$): Input is string $\alpha \in S_{n,j}$, compute index $x$ of $\alpha$ in $S_{n,j}$ in polynomial time in $n$.

2. **DecodeBinomCoeff**($x$): Input index $x \in \{1, \ldots, \binom{n}{j}\}$. Output $x$th string $\alpha$ in $S_{n,j}$, in time $O(\text{polylog } n + n)$. 
Extracting randomness

Theorem
Consider a coin that comes up heads with probability $p > 1/2$. For any constant $\delta > 0$ and for $n$ sufficiently large: [ (A)]

One can extract, from an input of a sequence of $n$ flips, an output sequence of $(1 - \delta)n \mathbb{H}(p)$ (unbiased) independent random bits.

One can not extract more than $n \mathbb{H}(p)$ bits from such a sequence.
Proof...

1. There are \( \binom{n}{j} \) input strings with exactly \( j \) heads.
2. each has probability \( p^j (1 - p)^{n-j} \).
3. map string \( s \) like that to index number in the set
   \[ S_j = \{1, \ldots, \binom{n}{j}\}. \]
4. Given that input string \( s \) has \( j \) ones (out of \( n \) bits)
   defines a uniform distribution on \( S_{n,j} \).
5. \( x \leftarrow \text{EncodeBinomCoeff}(s) \)
6. \( x \) uniform distributed in \( \{1, \ldots, N\} \), \( N = \binom{n}{j} \).
7. Seen in previous lecture...
8. ... extract in expectation, \( \lfloor \log N \rfloor - 1 \) bits from
   uniform random variable in the range \( 1, \ldots, N \).
Proof...

1. There are \( \binom{n}{j} \) input strings with exactly \( j \) heads.
2. each has probability \( p^j(1 - p)^{n-j} \).
3. map string \( s \) like that to index number in the set 
   \[ S_j = \{1, \ldots, \binom{n}{j}\} \] .
4. Given that input string \( s \) has \( j \) ones (out of \( n \) bits) 
   defines a uniform distribution on \( S_{n,j} \).
5. \( x \leftarrow \text{EncodeBinomCoeff}(s) \)
6. \( x \) uniform distributed in \( \{1, \ldots, N\} \), \( N = \binom{n}{j} \).
7. Seen in previous lecture...
8. ... extract in expectation, \( \lfloor \log N \rfloor - 1 \) bits from 
   uniform random variable in the range \( 1, \ldots, N \).
Proof...

1. There are \((n \choose j)\) input strings with exactly \(j\) heads.

2. Each has probability \(p^j(1 - p)^{n-j}\).

3. Map string \(s\) like that to index number in the set \(S_j = \{1, \ldots, (n \choose j)\}\).

4. Given that input string \(s\) has \(j\) ones (out of \(n\) bits) defines a uniform distribution on \(S_{n,j}\).

5. \(x \leftarrow \text{EncodeBinomCoeff}(s)\)

6. \(x\) uniform distributed in \(\{1, \ldots, N\}\), \(N = (n \choose j)\).

7. Seen in previous lecture...

8. ... extract in expectation, \([\lg N] - 1\) bits from uniform random variable in the range \(1, \ldots, N\).
Proof...

1. There are $\binom{n}{j}$ input strings with exactly $j$ heads.
2. Each has probability $p^j(1 - p)^{n-j}$.
3. Map string $s$ like that to index number in the set $S_j = \left\{1, \ldots, \binom{n}{j}\right\}$.
4. Given that input string $s$ has $j$ ones (out of $n$ bits) defines a uniform distribution on $S_{n,j}$.
5. $x \leftarrow \text{EncodeBinomCoeff}(s)$
6. $x$ uniform distributed in $\{1, \ldots, N\}$, $N = \binom{n}{j}$.
7. Seen in previous lecture...
8. ... extract in expectation, $\lfloor \log N \rfloor - 1$ bits from uniform random variable in the range $1, \ldots, N$. 
Proof...

1. There are $\binom{n}{j}$ input strings with exactly $j$ heads.
2. each has probability $p^j(1 - p)^{n-j}$.
3. map string $s$ like that to index number in the set $S_j = \{1, \ldots, \binom{n}{j}\}$.
4. Given that input string $s$ has $j$ ones (out of $n$ bits) defines a uniform distribution on $S_{n,j}$.
5. $x \leftarrow \text{EncodeBinomCoeff}(s)$
6. $x$ uniform distributed in $\{1, \ldots, N\}$, $N = \binom{n}{j}$.
7. Seen in previous lecture...
8. ... extract in expectation, $\lceil \lg N \rceil - 1$ bits from uniform random variable in the range $1, \ldots, N$. 
Proof...

1. There are \( \binom{n}{j} \) input strings with exactly \( j \) heads.
2. each has probability \( p^j(1 - p)^{n-j} \).
3. map string \( s \) like that to index number in the set \( S_j = \{1, \ldots, \binom{n}{j}\} \).
4. Given that input string \( s \) has \( j \) ones (out of \( n \) bits) defines a uniform distribution on \( S_{n,j} \).
5. \( x \leftarrow \text{EncodeBinomCoeff}(s) \)
6. \( x \) uniform distributed in \( \{1, \ldots, N\} \), \( N = \binom{n}{j} \).
7. Seen in previous lecture...
8. ... extract in expectation, \( \lfloor \log N \rfloor - 1 \) bits from uniform random variable in the range \( 1, \ldots, N \).
Proof...

1. There are \( \binom{n}{j} \) input strings with exactly \( j \) heads.
2. each has probability \( p^j(1 - p)^{n-j} \).
3. map string \( s \) like that to index number in the set \( S_j = \{1, \ldots, \binom{n}{j}\} \).
4. Given that input string \( s \) has \( j \) ones (out of \( n \) bits) defines a uniform distribution on \( S_{n,j} \).
5. \( x \leftarrow \text{EncodeBinomCoeff}(s) \)
6. \( x \) uniform distributed in \( \{1, \ldots, N\} \), \( N = \binom{n}{j} \).
7. Seen in previous lecture...
8. ... extract in expectation, \( \lfloor \lg N \rfloor - 1 \) bits from uniform random variable in the range \( 1, \ldots, N \).
Proof...

1. There are \( \binom{n}{j} \) input strings with exactly \( j \) heads.
2. Each has probability \( p^j(1 - p)^{n-j} \).
3. Map string \( s \) like that to index number in the set \( S_j = \{1, \ldots, \binom{n}{j}\} \).
4. Given that input string \( s \) has \( j \) ones (out of \( n \) bits) defines a uniform distribution on \( S_{n,j} \).
5. \( x \leftarrow \text{EncodeBinomCoeff}(s) \)
6. \( x \) uniform distributed in \( \{1, \ldots, N\} \), \( N = \binom{n}{j} \).
7. Seen in previous lecture...
8. ... extract in expectation, \( \lfloor \lg N \rfloor - 1 \) bits from uniform random variable in the range \( 1, \ldots, N \).
Proof...

1. There are \( \binom{n}{j} \) input strings with exactly \( j \) heads.
2. each has probability \( p^j(1 - p)^{n-j} \).
3. map string \( s \) like that to index number in the set \( S_j = \{ 1, \ldots, \binom{n}{j} \} \).
4. Given that input string \( s \) has \( j \) ones (out of \( n \) bits)
   defines a uniform distribution on \( S_{n,j} \).
5. \( x \leftarrow \text{EncodeBinomCoeff}(s) \)
6. \( x \) uniform distributed in \( \{ 1, \ldots, N \} \), \( N = \binom{n}{j} \).
7. Seen in previous lecture...
8. ... extract in expectation, \( \lfloor \log N \rfloor - 1 \) bits from
   uniform random variable in the range \( 1, \ldots, N \).
Exciting proof continued...

1. \( Z \): random variable: number of heads in input string \( s \).

2. \( B \): number of random bits extracted.

\[
E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k],
\]

3. Know: \( E[B \mid Z = k] \geq \left\lfloor \lg \left( \binom{n}{k} \right) \right\rfloor - 1 \).

4. \( \varepsilon < p - 1/2 \): sufficiently small constant.

5. \( n(p - \varepsilon) \leq k \leq n(p + \varepsilon) \):

\[
\binom{n}{k} \geq \binom{n}{\lfloor n(p + \varepsilon) \rfloor} \geq \frac{2^n \mathbb{H}(p + \varepsilon)}{n + 1},
\]
Exciting proof continued...

1. **Z**: random variable: number of heads in input string *s*.

2. **B**: number of random bits extracted.

   \[
   \mathbb{E}[B] = \sum_{k=0}^{n} \Pr[Z = k] \mathbb{E}[B \mid Z = k],
   \]

3. Know: \( \mathbb{E}[B \mid Z = k] \geq \left\lfloor \log \left( \binom{n}{k} \right) \right\rfloor - 1 \).

4. \( \varepsilon < p - 1/2 \): sufficiently small constant.

5. \( n(p - \varepsilon) \leq k \leq n(p + \varepsilon) \):

   \[
   \binom{n}{k} \geq \left\lfloor n(p + \varepsilon) \right\rfloor \geq \frac{2^{nH(p+\varepsilon)}}{n+1},
   \]
Exciting proof continued...

1. $Z$: random variable: number of heads in input string $s$.
2. $B$: number of random bits extracted.

$$
E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k],
$$

3. Know: $E[B \mid Z = k] \geq \left\lceil \log \left( \frac{n}{k} \right) \right\rceil - 1$.
4. $\varepsilon < p - 1/2$: sufficiently small constant.
5. $n(p - \varepsilon) \leq k \leq n(p + \varepsilon)$:

$$
\binom{n}{k} \geq \left\lfloor n(p + \varepsilon) \right\rfloor \geq \frac{2^{nH(p+\varepsilon)}}{n+1},
$$
Exciting proof continued...

1. **Z**: random variable: number of heads in input string \( s \).
2. **B**: number of random bits extracted.
   \[
   \mathbb{E}[B] = \sum_{k=0}^{n} \Pr[Z = k] \mathbb{E}[B \mid Z = k],
   \]
3. Know: \( \mathbb{E}[B \mid Z = k] \geq \left\lfloor \lg \binom{n}{k} \right\rfloor - 1. \)
4. \( \varepsilon < p - 1/2 \): sufficiently small constant.
5. \( n(p - \varepsilon) \leq k \leq n(p + \varepsilon) \):
   \[
   \binom{n}{k} \geq \binom{n}{\left\lfloor n(p + \varepsilon) \right\rfloor} \geq \frac{2^{n\mathbb{H}(p + \varepsilon)}}{n + 1},
   \]
Exciting proof continued...

1. \( Z \): random variable: number of heads in input string \( s \).
2. \( B \): number of random bits extracted.

\[
E[B] = \sum_{k=0}^{n} \Pr[Z = k] \, E\left[B \mid Z = k\right],
\]

3. Know: \( E\left[B \mid Z = k\right] \geq \left\lfloor \log \left( \binom{n}{k} \right) \right\rfloor - 1 \).
4. \( \varepsilon < p - 1/2 \): sufficiently small constant.
5. \( n(p - \varepsilon) \leq k \leq n(p + \varepsilon) \):

\[
\binom{n}{k} \geq \left( \lfloor n(p + \varepsilon) \rfloor \right) \geq \frac{2^{n\mathbb{H}(p+\varepsilon)}}{n + 1},
\]
Exciting proof continued...

1. $Z$: random variable: number of heads in input string $s$.

2. $B$: number of random bits extracted.

$$E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k],$$

3. Know: $E[B \mid Z = k] \geq \left\lfloor \lg \left( \frac{n}{k} \right) \right\rfloor - 1$.

4. $\varepsilon < p - 1/2$: sufficiently small constant.

5. $n(p - \varepsilon) \leq k \leq n(p + \varepsilon)$:

$$\binom{n}{k} \geq \binom{n}{\lfloor n(p + \varepsilon) \rfloor} \geq \frac{2^{n\mathbb{H}(p+\varepsilon)}}{n + 1},$$
Exciting proof continued...

1. \( Z \): random variable: number of heads in input string \( s \).
2. \( B \): number of random bits extracted.

\[
E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k],
\]

3. Know: \( E[B \mid Z = k] \geq \left\lfloor \log \left( \frac{n}{k} \right) \right\rfloor - 1 \).
4. \( \varepsilon < p - 1/2 \): sufficiently small constant.
5. \( n(p - \varepsilon) \leq k \leq n(p + \varepsilon) \):

\[
\binom{n}{k} \geq \binom{n}{\lfloor n(p + \varepsilon) \rfloor} \geq \frac{2^{nH(p + \varepsilon)}}{n + 1},
\]
Super exciting proof continued...

\[ E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k]. \]

\[ E[B] \geq \sum_{k=\lfloor n(p-\varepsilon) \rfloor}^{\lfloor n(p+\varepsilon) \rfloor} \Pr[Z = k] \left( \left\lfloor \lg \left( \frac{n}{k} \right) \right\rfloor - 1 \right) \]

\[ \geq \sum_{k=\lfloor n(p-\varepsilon) \rfloor}^{\lfloor n(p+\varepsilon) \rfloor} \Pr[Z = k] \left( \lg \frac{2^{nH(p+\varepsilon)}}{n + 1} - 2 \right) \]

\[ = \left( nH(p + \varepsilon) - \lg(n + 1) - 2 \right) \Pr[|Z - np| \leq \varepsilon n] \]

\[ \geq \left( nH(p + \varepsilon) - \lg(n + 1) - 2 \right) \left( 1 - 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right) \right), \]
Super exciting proof continued...

\[ \mathbb{E}[B] = \sum_{k=0}^{n} \Pr[Z = k] \mathbb{E}[B \mid Z = k]. \]

\[ \mathbb{E}[B] \geq \sum_{k=\lfloor n(p-\varepsilon) \rfloor}^{\lfloor n(p+\varepsilon) \rfloor} \Pr[Z = k] \left( \left\lceil \lg \binom{n}{k} \right\rceil - 1 \right) \]

\[ \geq \sum_{k=\lfloor n(p-\varepsilon) \rfloor}^{\lfloor n(p+\varepsilon) \rfloor} \Pr[Z = k] \left( \lg \frac{2^{n\mathbb{H}(p+\varepsilon)}}{n + 1} - 2 \right) \]

\[ = \left( n\mathbb{H}(p + \varepsilon) - \lg(n + 1) - 2 \right) \Pr[|Z - np| \leq \varepsilon n] \]

\[ \geq \left( n\mathbb{H}(p + \varepsilon) - \lg(n + 1) - 2 \right) \left( 1 - 2 \exp\left(-\frac{n\varepsilon^2}{4p}\right) \right), \]
Super exciting proof continued...

\[
\mathbb{E}[B] = \sum_{k=0}^{n} \Pr[Z = k] \mathbb{E}[B \mid Z = k].
\]

\[
\mathbb{E}[B] \geq \sum_{k=[n(p-\varepsilon)]}^{[n(p+\varepsilon)]} \Pr[Z = k] \mathbb{E}[B \mid Z = k]
\]

\[
\geq \sum_{k=[n(p-\varepsilon)]}^{[n(p+\varepsilon)]} \Pr[Z = k] \left( \left\lfloor \frac{\log(n)}{k} \right\rfloor - 1 \right)
\]

\[
\geq \sum_{k=[n(p-\varepsilon)]}^{[n(p+\varepsilon)]} \Pr[Z = k] \left( \log \frac{2^nH(p+\varepsilon)}{n+1} - 2 \right)
\]

\[
\leq \left( nH(p + \varepsilon) - \log(n+1) - 2 \right) \Pr[|Z - np| \leq \varepsilon n]
\]

\[
\geq \left( nH(p + \varepsilon) - \log(n+1) - 2 \right) \left( 1 - 2 \exp\left( -\frac{n\varepsilon^2}{4p} \right) \right),
\]
Super exciting proof continued...

\[ E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k]. \]

\[ E[B] \geq \sum_{k=[n(p-\varepsilon)]}^{[n(p+\varepsilon)]} \Pr[Z = k] E[B \mid Z = k] \]

\[ \geq \sum_{k=[n(p-\varepsilon)]}^{[n(p+\varepsilon)]} \Pr[Z = k] \left( \left\lceil \log \left( \frac{n}{k} \right) \right\rceil - 1 \right) \]

\[ \geq \sum_{k=[n(p-\varepsilon)]}^{[n(p+\varepsilon)]} \Pr[Z = k] \left( \log \frac{2n H(p+\varepsilon)}{n+1} - 2 \right) \]

\[ = \left( n H(p+\varepsilon) - \log(n+1) - 2 \right) \Pr[|Z - np| \leq \varepsilon n] \]

\[ \geq \left( n H(p+\varepsilon) - \log(n+1) - 2 \right) \left( 1 - 2 \exp\left( -\frac{n \varepsilon^2}{4p} \right) \right), \]
Super exciting proof continued...

\[
\begin{align*}
\mathbb{E}[B] &= \sum_{k=0}^{n} \Pr[Z = k] \mathbb{E}[B \mid Z = k]. \\
\mathbb{E}[B] &\geq \sum_{k=\lceil n(p-\varepsilon) \rceil}^{\lfloor n(p+\varepsilon) \rfloor} \Pr[Z = k] \mathbb{E}[B \mid Z = k] \\
&\geq \sum_{k=\lceil n(p-\varepsilon) \rceil}^{\lfloor n(p+\varepsilon) \rfloor} \Pr[Z = k] \left( \left\lfloor \log \left( \frac{n}{k} \right) \right\rfloor - 1 \right) \\
&\geq \sum_{k=\lceil n(p-\varepsilon) \rceil}^{\lfloor n(p+\varepsilon) \rfloor} \Pr[Z = k] \left( \log \frac{2^{n \mathbb{H}(p+\varepsilon)}}{n + 1} - 2 \right) \\
&= \left( n \mathbb{H}(p + \varepsilon) - \log(n + 1) - 2 \right) \Pr[|Z - np| \leq \varepsilon n] \\
&\geq \left( n \mathbb{H}(p + \varepsilon) - \log(n + 1) - 2 \right) \left( 1 - 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right) \right),
\end{align*}
\]
Super exciting proof continued...

\[
E[B] = \sum_{k=0}^{n} \Pr[Z = k] \cdot E[B \mid Z = k].
\]

\[
E[B] \geq \sum_{k=[n(p-\varepsilon)]}^{[n(p+\varepsilon)]} \Pr[Z = k] \cdot E[B \mid Z = k]
\]

\[
\geq \sum_{k=[n(p-\varepsilon)]}^{[n(p+\varepsilon)]} \Pr[Z = k] \left( \left\lfloor \log \binom{n}{k} \right\rfloor - 1 \right)
\]

\[
\geq \sum_{k=[n(p-\varepsilon)]}^{[n(p+\varepsilon)]} \Pr[Z = k] \left( \log \frac{2^{nH(p+\varepsilon)}}{n + 1} - 2 \right)
\]

\[
= \left( nH(p + \varepsilon) - \log(n + 1) - 2 \right) \Pr[|Z - np| \leq \varepsilon n]
\]

\[
\geq \left( nH(p + \varepsilon) - \log(n + 1) - 2 \right) \left( 1 - 2 \exp\left(-\frac{n\varepsilon^2}{4p}\right) \right),
\]

since \( \mu = E[Z] = np \) and \( \Pr[|Z - np| \geq \varepsilon n] \leq 2 \exp\left(-np(\varepsilon/2)^2\right) \), by the Chernoff inequality.
Super exciting proof continued...

\[
\begin{align*}
\mathbb{E}[B] &= \sum_{k=0}^{n} \Pr[Z = k] \mathbb{E}[B \mid Z = k] . \\
\mathbb{E}[B] &\geq \sum_{k=\lfloor n(p-\epsilon) \rfloor}^{\lfloor n(p+\epsilon) \rfloor} \Pr[Z = k] \left( \left\lfloor \log \left( \frac{n}{k} \right) \right\rfloor - 1 \right) \\
&\geq \sum_{k=\lfloor n(p-\epsilon) \rfloor}^{\lfloor n(p+\epsilon) \rfloor} \Pr[Z = k] \left( \log \frac{2^{n\mathbb{H}(p+\epsilon)}}{n + 1} - 2 \right) \\
&= \left( n\mathbb{H}(p + \epsilon) - \log(n + 1) - 2 \right) \Pr[|Z - np| \leq \epsilon n] \\
&\geq \left( n\mathbb{H}(p + \epsilon) - \log(n + 1) - 2 \right) \left( 1 - 2 \exp \left( -\frac{n\epsilon^2}{4p} \right) \right),
\end{align*}
\]
Hyper super exciting proof continued...

1. Fix $\varepsilon > 0$, such that
   \[ H(p + \varepsilon) > (1 - \delta/4)H(p), \]  
   $p$ is fixed.

2. \[ \iff nH(p) = \Omega(n), \]  

3. For $n$ sufficiently large:
   \[ -\lg(n + 1) \geq -\frac{\delta}{10} nH(p). \]  

4. Also, \[ 2 \exp\left(-\frac{n\varepsilon^2}{4p}\right) \leq \frac{\delta}{10}. \]  

5. For $n$ large enough;
   \[ E[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right)nH(p)\left(1 - \frac{\delta}{10}\right) \]  
   \[ \geq (1 - \delta)nH(p), \]
Hyper super exciting proof continued...

1. Fix $\epsilon > 0$, such that
   \[ H(p + \epsilon) > (1 - \delta/4)H(p), \] $p$ is fixed.

2. $\implies nH(p) = \Omega(n)$,

3. For $n$ sufficiently large:
   \[ -\log(n + 1) \geq -\frac{\delta}{10} nH(p). \]

4. ... also \[ 2 \exp\left(-\frac{n\epsilon^2}{4p}\right) \leq \frac{\delta}{10}. \]

5. For $n$ large enough;

   \[
   E[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right) nH(p) \left(1 - \frac{\delta}{10}\right)
   \geq (1 - \delta) nH(p),
   \]
Hyper super exciting proof continued...

1. Fix $\varepsilon > 0$, such that
   \[ H(p + \varepsilon) > (1 - \frac{\delta}{4})H(p), \text{ } p \text{ is fixed}. \]

2. $\implies nH(p) = \Omega(n),$

3. For $n$ sufficiently large:
   \[ -\log(n + 1) \geq -\frac{\delta}{10}nH(p). \]

4. ... also $2 \exp\left(-\frac{n\varepsilon^2}{4p}\right) \leq \frac{\delta}{10}.$

5. For $n$ large enough;

\[
E[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right)nH(p)\left(1 - \frac{\delta}{10}\right) \\
\geq (1 - \delta)nH(p),
\]
Hyper super exciting proof continued...

1. Fix $\varepsilon > 0$, such that
   \[ \mathbb{H}(p + \varepsilon) > (1 - \frac{\delta}{4})\mathbb{H}(p), \] 
   $p$ is fixed.

2. $\implies n\mathbb{H}(p) = \Omega(n),$

3. For $n$ sufficiently large:
   \[ -\lg(n + 1) \geq -\frac{\delta}{10} n\mathbb{H}(p). \]

4. ... also
   \[ 2 \exp\left(-\frac{n\varepsilon^2}{4p}\right) \leq \frac{\delta}{10}. \]

5. For $n$ large enough;

   \[ E[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right) n\mathbb{H}(p) \left(1 - \frac{\delta}{10}\right) \]
   \[ \geq (1 - \delta) n\mathbb{H}(p), \]
1. Fix $\varepsilon > 0$, such that $\mathbb{H}(p + \varepsilon) > (1 - \delta/4)\mathbb{H}(p)$, $p$ is fixed.
2. $\implies n\mathbb{H}(p) = \Omega(n)$,
3. For $n$ sufficiently large:
   $-\lg(n + 1) \geq -\frac{\delta}{10} n\mathbb{H}(p)$.
4. ... also $2^{\exp\left(-\frac{n\varepsilon^2}{4p}\right)} \leq \frac{\delta}{10}$.
5. For $n$ large enough;

$$
E[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right) n\mathbb{H}(p) \left(1 - \frac{\delta}{10}\right) \\
\geq (1 - \delta) n\mathbb{H}(p),
$$
Hyper super duper exciting proof continued...

1. Need to prove upper bound.

2. If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$.

3. All sequences of length $|y|$ have equal probability to be generated (by definition).

4. $2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1$.

5. $\implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x])$

6. $\mathbb{E}[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$
   \[
   \leq \sum_x \Pr[X = x] \log \frac{1}{\Pr[X = x]} = \mathbb{H}(X).
   \]
Hyper super duper exciting proof continued...

1. Need to prove upper bound.

2. If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$.

3. All sequences of length $|y|$ have equal probability to be generated (by definition).

4. $2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1.$

5. $\implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x])$

6. $E[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$  
   $\leq \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X = x]} = \mathbb{H}(X).$
Hyper super duper exciting proof continued...

1. Need to prove upper bound.
2. If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$.
3. All sequences of length $|y|$ have equal probability to be generated (by definition).
4. $2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1.$
5. $\implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x])$
6. $E[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$
   $\leq \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X = x]} = \mathbb{H}(X).$
Hyper super duper exciting proof continued...

1. Need to prove upper bound.
2. If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$.
3. All sequences of length $|y|$ have equal probability to be generated (by definition).
4. $2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1.$
5. $\implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x])$
6. $\mathbb{E}[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$
   $\leq \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X=x]} = \mathbb{H}(X)$. 
Hyper super duper exciting proof continued...

1. Need to prove upper bound.
2. If input sequence \( x \) has probability \( \Pr[X = x] \), then \( y = \text{Ext}(x) \) has probability to be generated \( \geq \Pr[X = x] \).
3. All sequences of length \(|y|\) have equal probability to be generated (by definition).
4. \[ 2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1. \]
5. \[ \implies |\text{Ext}(x)| \leq \log(1/\Pr[X = x]) \]
6. \[ \mathbb{E}[B] = \sum_x \Pr[X = x] |\text{Ext}(x)| \]
   \[ \leq \sum_x \Pr[X = x] \log \frac{1}{\Pr[X = x]} = \mathbb{H}(X). \]
Hyper super duper exciting proof continued...

1. Need to prove upper bound.
2. If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$.
3. All sequences of length $|y|$ have equal probability to be generated (by definition).
4. $2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1.$
5. $\implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x])$
6. $E[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$
    $\leq \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X = x]} = H(X)$. 
Hyper super duper exciting proof continued...

1. Need to prove upper bound.
2. If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$.
3. All sequences of length $|y|$ have equal probability to be generated (by definition).
4. $2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1$.
5. $\Rightarrow |\text{Ext}(x)| \leq \log(1/\Pr[X = x])$
6. $\mathbb{E}[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$
   $\leq \sum_x \Pr[X = x] \log \frac{1}{\Pr[X = x]} = \mathbb{H}(X)$. 
Hyper super duper exciting proof continued...

1. Need to prove upper bound.
2. If input sequence \( x \) has probability \( \Pr[X = x] \), then \( y = \text{Ext}(x) \) has probability to be generated \( \geq \Pr[X = x] \).
3. All sequences of length \(|y|\) have equal probability to be generated (by definition).
4. \( 2^{\text{Ext}(x)} \) \( \Pr[X = x] \leq \frac{1}{2^{\text{Ext}(x)}} \Pr[y = \text{Ext}(x)] \leq 1 \).
5. \( \implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x]) \)
6. \( E[B] = \sum_x \Pr[X = x] |\text{Ext}(x)| \leq \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X = x]} = \mathbb{H}(X). \)
Hyper super duper exciting proof continued...

1. Need to prove upper bound.
2. If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$.
3. All sequences of length $|y|$ have equal probability to be generated (by definition).
4. $2^{\text{Ext}(x)} \Pr[X = x] \leq 2^{\text{Ext}(x)} \Pr[y = \text{Ext}(x)] \leq 1.$
5. $\implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x])$
6. $\mathbb{E}[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$
   $\leq \sum_x \Pr[X = x] \log \frac{1}{\Pr[X = x]} = \mathbb{H}(X)$. 
Part III

Coding: Shannon’s Theorem
Shannon’s Theorem

Definition

1. *binary symmetric channel* with parameter $p$
2. sequence of bits $x_1, x_2, \ldots$, an
3. output: $y_1, y_2, \ldots$, a sequence of bits such that...
4. $\Pr[x_i = y_i] = 1 - p$ independently for each $i$. 
Shannon’s Theorem

Definition

1. *binary symmetric channel* with parameter $p$
2. sequence of bits $x_1, x_2, \ldots$, an
3. output: $y_1, y_2, \ldots$, a sequence of bits such that...
4. $\Pr[x_i = y_i] = 1 - p$ independently for each $i$. 
Shannon’s Theorem

Definition

1. **binary symmetric channel** with parameter \( p \)
2. sequence of bits \( x_1, x_2, \ldots \), an
3. output: \( y_1, y_2, \ldots \),
   a sequence of bits such that...
4. \( \Pr[x_i = y_i] = 1 - p \) independently for each \( i \).
Shannon’s Theorem

Definition

1. *binary symmetric channel* with parameter $p$
2. sequence of bits $x_1, x_2, \ldots$, an
3. output: $y_1, y_2, \ldots$, a sequence of bits such that...
4. $\Pr[x_i = y_i] = 1 - p$ independently for each $i$. 
Encoding/decoding with noise

Definition

1. \((k, n)\) **encoding function**
   \[
   \text{Enc} : \{0, 1\}^k \rightarrow \{0, 1\}^n
   \]
   takes as input a sequence of \(k\) bits and outputs a sequence of \(n\) bits.

2. \((k, n)\) **decoding function**
   \[
   \text{Dec} : \{0, 1\}^n \rightarrow \{0, 1\}^k
   \]
   takes as input a sequence of \(n\) bits and outputs a sequence of \(k\) bits.
Encoding/decoding with noise

Definition

1. \((k, n)\) encoding function
   \[
   \text{Enc} : \{0, 1\}^k \rightarrow \{0, 1\}^n
   \]
   takes as input a sequence of \(k\) bits and outputs a sequence of \(n\) bits.

2. \((k, n)\) decoding function
   \[
   \text{Dec} : \{0, 1\}^n \rightarrow \{0, 1\}^k
   \]
   takes as input a sequence of \(n\) bits and outputs a sequence of \(k\) bits.
Claude Elwood Shannon

Claude Elwood Shannon (April 30, 1916 - February 24, 2001), an American electrical engineer and mathematician, has been called “the father of information theory”. His master thesis was how to building boolean circuits for any boolean function.
Shannon’s theorem (1948)

Theorem (Shannon’s theorem)

For a binary symmetric channel with parameter $p < 1/2$ and for any constants $\delta, \gamma > 0$, where $n$ is sufficiently large, the following holds: [(i)]

For an $k \leq n(1 - \mathbb{H}(p) - \delta)$ there exists $(k, n)$ encoding and decoding functions such that the probability the receiver fails to obtain the correct message is at most $\gamma$ for every possible $k$-bit input messages.

There are no $(k, n)$ encoding and decoding functions with $k \geq n(1 - \mathbb{H}(p) + \delta)$ such that the probability of decoding correctly is at least $\gamma$ for a $k$-bit input message.
When the sender sends a string...

\[ S = s_1 s_2 \ldots s_n \]
When the sender sends a string...

\[ S = s_1 s_2 \ldots s_n \]
When the sender sends a string...

\[ S = s_1 s_2 \ldots s_n \]
When the sender sends a string...

\[ S = s_1 s_2 \ldots s_n \]
When the sender sends a string...

\[ S = s_1 s_2 \ldots s_n \]

One ring to rule them all!
Some intuition...

1. senders sent string \( S = s_1 s_2 \ldots s_n \).
2. receiver got string \( T = t_1 t_2 \ldots t_n \).
3. \( p = \Pr[t_i \neq s_i] \), for all \( i \).
4. \( U \): Hamming distance between \( S \) and \( T \):
   \[ U = \sum_i [s_i \neq t_i] \].
5. By assumption: \( \mathbb{E}[U] = pn \), and \( U \) is a binomial variable.
6. By Chernoff inequality:
   \[ U \in [(1 - \delta)np, (1 + \delta)np] \] with high probability, where \( \delta \) is tiny constant.
7. \( T \) is in a ring \( R \) centered at \( S \), with inner radius \( (1 - \delta)np \) and outer radius \( (1 + \delta)np \).
8. This ring has
Some intuition...

1. senders sent string $S = s_1 s_2 \ldots s_n$.
2. receiver got string $T = t_1 t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$:
   $$U = \sum_i [s_i \neq t_i].$$
5. By assumption: $\mathbb{E}[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality:
   $$U \in [(1 - \delta)np, (1 + \delta)np]$$
   with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has
Some intuition...

1. senders sent string $S = s_1 s_2 \ldots s_n$.
2. receiver got string $T = t_1 t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$:
   $$U = \sum_i [s_i \neq t_i].$$
5. By assumption: $E[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality:
   $$U \in [(1 - \delta)np, (1 + \delta)np]$$
   with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has
Some intuition...

1. senders sent string $S = s_1s_2 \ldots s_n$.
2. receiver got string $T = t_1t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$:
   $U = \sum_i[s_i \neq t_i]$.
5. By assumption: $\mathbb{E}[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality:
   $U \in [(1 - \delta)np, (1 + \delta)np]$ with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has
Some intuition...

1. senders sent string $S = s_1s_2 \ldots s_n$.
2. receiver got string $T = t_1t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$:
   $$U = \sum_i[s_i \neq t_i].$$
5. By assumption: $\mathbb{E}[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality:
   $$U \in [(1 - \delta)np, (1 + \delta)np]$$
   with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has
Some intuition...

1. senders sent string \( S = s_1 s_2 \ldots s_n \).
2. receiver got string \( T = t_1 t_2 \ldots t_n \).
3. \( p = \Pr[t_i \neq s_i] \), for all \( i \).
4. \( U \): Hamming distance between \( S \) and \( T \):
   \[ U = \sum_i [s_i \neq t_i] . \]
5. By assumption: \( \mathbb{E}[U] = pn \), and \( U \) is a binomial variable.
6. By Chernoff inequality:
   \[ U \in [(1 - \delta)np, (1 + \delta)np] \]
   with high probability, where \( \delta \) is tiny constant.
7. \( T \) is in a ring \( R \) centered at \( S \), with inner radius \( (1 - \delta)np \) and outer radius \( (1 + \delta)np \).
8. This ring has
Some intuition...

1. senders sent string $S = s_1s_2 \ldots s_n$.
2. receiver got string $T = t_1t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$: $U = \sum_i[s_i \neq t_i]$.
5. By assumption: $E[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality: $U \in [(1 - \delta)np, (1 + \delta)np]$ with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has
Some intuition...

1. senders sent string $S = s_1 s_2 \ldots s_n$.
2. receiver got string $T = t_1 t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$: $U = \sum_i[s_i \neq t_i]$.
5. By assumption: $E[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality: $U \in [(1 - \delta)np, (1 + \delta)np]$ with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has
Many rings for many codewords...
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.
2. If every word in the hypercube would be covered...
3. ... use $2^n$ codewords $\implies \kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{nH((1+\delta)p)}} \approx 2^{n(1-H((1+\delta)p))}$.

4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.
5. Map $i$th string in $\{0, 1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.
6. If send $C_i \implies$ receiver gets a string in $R_i$.
7. Decoding is easy - find the ring $R_i$ containing the received string, take its center string $C_i$, and output the original string it was mapped to.

8. How many bits? $k = \lfloor \log_2 \kappa \rfloor \approx n(1-H((1+\delta)p))$. 
Some more intuition...

1. Pick as many disjoint rings as possible: \( R_1, \ldots, R_\kappa \).
2. If every word in the hypercube would be covered...
3. ... use \( 2^n \) codewords \( \implies \kappa \geq \)

\[
\kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{nH((1+\delta)p)}} \approx 2^n(1-H((1+\delta)p)).
\]

4. Consider all possible strings of length \( k \) such that \( 2^k \leq \kappa \).
5. Map \( i \)th string in \( \{0,1\}^k \) to the center \( C_i \) of the \( i \)th ring \( R_i \).
6. If send \( C_i \) \( \implies \) receiver gets a string in \( R_i \).
7. Decoding is easy - find the ring \( R_i \) containing the received string, take its center string \( C_i \), and output the original string it was mapped to.

8. How many bits?

\[
k = \left\lfloor \log_2 \kappa \right\rfloor = n \left(1-H((1+\delta)p))\right).\]
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.
2. If every word in the hypercube would be covered...
3. ... use $2^n$ codewords $\implies \kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^n H((1+\delta)p)} \approx 2^n (1 - H((1+\delta)p))$.

4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.
5. Map $i$th string in $\{0, 1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.
6. If send $C_i \implies$ receiver gets a string in $R_i$.
7. Decoding is easy - find the ring $R_i$ containing the received string, take its center string $C_i$, and...
Some more intuition...

1. Pick as many disjoint rings as possible: \( R_1, \ldots, R_\kappa \).

2. If every word in the hypercube would be covered...
3. ... use \( 2^n \) codewords \( \implies \kappa \geq \)

\[
\kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{nH((1+\delta)p)}} \approx 2^{n\left(1-\mathbb{H}((1+\delta)p)\right)}.
\]

4. Consider all possible strings of length \( k \) such that \( 2^k \leq \kappa \).
5. Map \( i \)th string in \( \{0, 1\}^k \) to the center \( C_i \) of the \( i \)th ring \( R_i \).
6. If send \( C_i \implies \) receiver gets a string in \( R_i \).
7. Decoding is easy - find the ring \( R_i \) containing the received string, take its center string \( C_i \), and...
Some more intuition...

1. Pick as many disjoint rings as possible: \( R_1, \ldots, R_\kappa \).
2. If every word in the hypercube would be covered...
3. ... use \( 2^n \) codewords \( \implies \kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{n \mathbb{H}((1+\delta)p)}} \approx 2^n(1-\mathbb{H}((1+\delta)p)). \)
4. Consider all possible strings of length \( k \) such that \( 2^k \leq \kappa \).
5. Map \( i \)th string in \( \{0, 1\}^k \) to the center \( C_i \) of the \( i \)th ring \( R_i \).
6. If send \( C_i \) \( \implies \) receiver gets a string in \( R_i \).
7. Decoding is easy - find the ring \( R_i \) containing the received string, take its center string \( C_i \), and output the original string it was mapped to.
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.
2. If every word in the hypercube would be covered...
3. ... use $2^n$ codewords $\implies \kappa \geq$

$$\kappa \geq \frac{2^n}{|R|} \geq 2 \cdot \frac{2^n}{2^{n\mathcal{H}((1+\delta)p)}} \approx 2^{n(1-\mathcal{H}((1+\delta)p))}.$$  

4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.
5. Map $i$th string in $\{0,1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.
6. If send $C_i$ $\implies$ receiver gets a string in $R_i$.
7. Decoding is easy - find the ring $R_i$ containing the received string, take its center string $C_i$, and output the original string it was mapped to.
8. How many bits?

$$k = \lfloor \log \kappa \rfloor = n(1-\mathcal{H}((1+\delta)p)) \approx n(1-\mathcal{H}(p)).$$
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.
2. If every word in the hypercube would be covered...
3. ... use $2^n$ codewords $\implies \kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{nH((1+\delta)p)}} \approx 2^{n(1-H((1+\delta)p))}$.
4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.
5. Map $i$th string in $\{0, 1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.
6. If send $C_i \implies$ receiver gets a string in $R_i$.
7. Decoding is easy - find the ring $R_i$ containing the received string, take its center string $C_i$, and output the original string it was mapped to.

8. How many bits?

$k = \lfloor \log \kappa \rfloor = n(1-H((1+\delta)p)) \approx n(1-H((1+\delta)p))$.
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.
2. If every word in the hypercube would be covered...
3. ... use $2^n$ codewords $\implies \kappa \geq$

$$\kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{nH((1+\delta)p)}} \approx 2^{n(1-H((1+\delta)p))}.$$ 

4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.
5. Map $i$th string in $\{0, 1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.
6. If send $C_i$ $\implies$ receiver gets a string in $R_i$.
7. Decoding is easy - find the ring $R_i$ containing the received string and output the string $C_i$. 
8. How many bits?

$$k = \lfloor \log_2 \kappa \rfloor = n(1-H((1+\delta)p)) \approx n(1-H((1+\delta)p)).$$
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.
2. If every word in the hypercube would be covered...
3. ... use $2^n$ codewords $\implies \kappa \geq$

$$\kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{(1+\delta)p}} \approx 2^{n(1-H((1+\delta)p))}.$$

4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.
5. Map $i$th string in $\{0,1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.
6. If send $C_i \implies$ receiver gets a string in $R_i$.
7. Decoding is easy - find the ring $R_i$ containing the received string and output the original string it was mapped to.
8. How many bits?

$$k = \lfloor \log \kappa \rfloor = n(1-H((1+\delta)p)) \approx n(1-\frac{\delta}{2p}) + \frac{\delta}{2p}.$$
What is wrong with the above?

1. Can not find such a large set of disjoint rings.
2. Reason is that when you pack rings (or balls) you are going to have wasted spaces around.
3. Overcome this: allow rings to overlap somewhat.
4. Makes things considerably more involved.
5. Details in class notes.
What is wrong with the above?

1. Can not find such a large set of disjoint rings.
2. Reason is that when you pack rings (or balls) you are going to have wasted spaces around.
3. Overcome this: allow rings to overlap somewhat.
4. Makes things considerably more involved.
5. Details in class notes.
What is wrong with the above?

1. Can not find such a large set of disjoint rings.
2. Reason is that when you pack rings (or balls) you are going to have wasted spaces around.
3. Overcome this: allow rings to overlap somewhat.
4. Makes things considerably more involved.
5. Details in class notes.
What is wrong with the above?

1. Can not find such a large set of disjoint rings.
2. Reason is that when you pack rings (or balls) you are going to have wasted spaces around.
3. Overcome this: allow rings to overlap somewhat.
4. Makes things considerably more involved.
5. Details in class notes.
What is wrong with the above?

1. Can not find such a large set of disjoint rings.
2. Reason is that when you pack rings (or balls) you are going to have wasted spaces around.
3. Overcome this: allow rings to overlap somewhat.
4. Makes things considerably more involved.
5. Details in class notes.
Notes
Notes