Entropy and Shannon's Theorem

Lecture 24

November 18, 2015

Part I

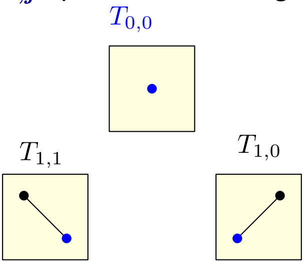
Entropy

Part II

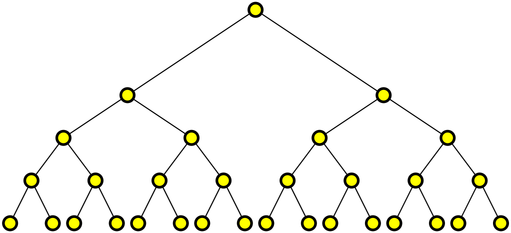
Extracting randomness

Storing all strings of length n and j bits on

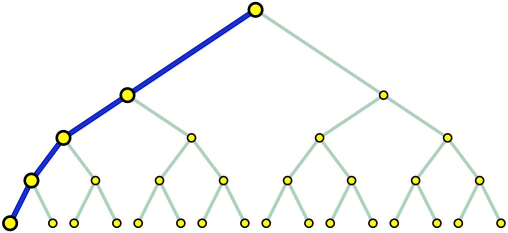
1. $S_{n,j}$: set of all strings of length n with j ones in them.
2. $T_{n,j}$: prefix tree storing all $S_{n,j}$.



Binary strings of length 4

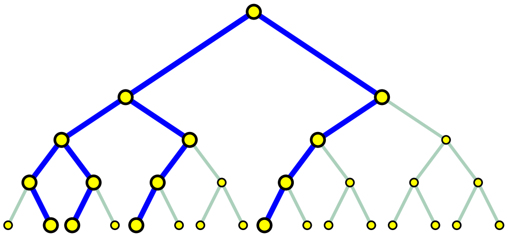


Binary strings of length 4



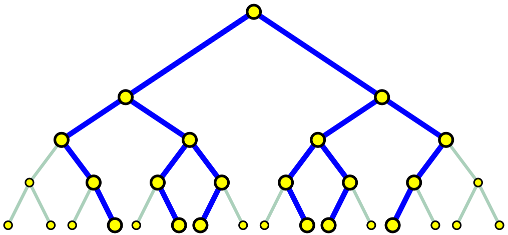
1. $S_{4,0} = \{0000\} \implies \#(0000) = 0$.
2.

Binary strings of length 4



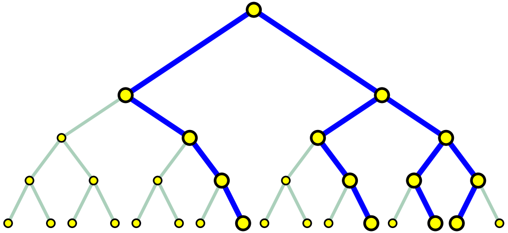
1. $S_{4,1} = \{0001, 0010, 0100, 1000\}$
   $\implies \#(0001) = 0$.
   $\#(0010) = 1$.
   $\#(0100) = 2$.
   $\#(1000) = 3$.
2.

Binary strings of length 4



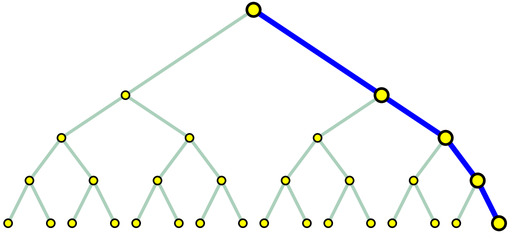
1. $S_{4,2} = \{0011, 0101, 0110, 1001, 1010, 1100\}$
   $\implies$
   $\#(0011) = 0.$
   $\#(0101) = 1.$
   $\#(0110) = 2.$
   $\#(1001) = 3.$
   $\#(1010) = 4.$
   $\#(1100) = 5.$

Binary strings of length 4



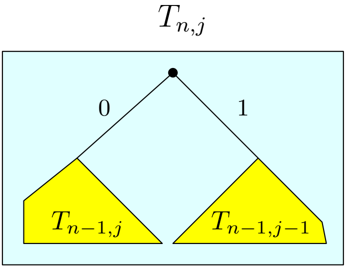
1. $S_{4,3} = \{0111, 1011, 1101, 1110\}$
   $\implies$
   $\#(0111) = 0$.
   $\#(1011) = 1$.
   $\#(1101) = 2$.
   $\#(1110) = 3$.
2.

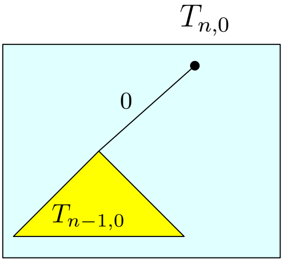
Binary strings of length 4

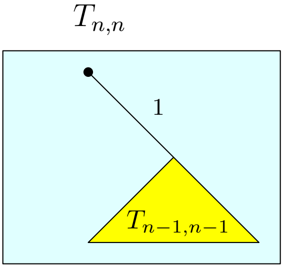


1. $S_{4,4} = \{1111\}$
   $\implies$
   $\#(1111) = 0$.
2.

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

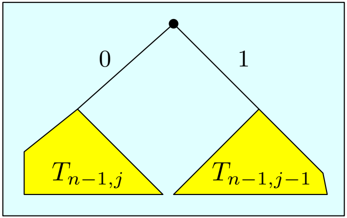






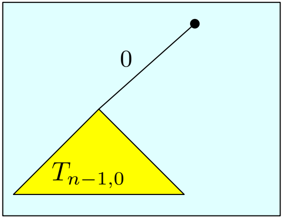
Prefix tree $\forall$ binary strings of length $\mathbf{n}$ with $\mathbf{j}$ ones

$T_{n,j}$

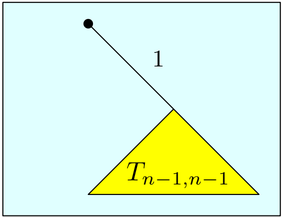


\# of leafs:
$|T_{n,j}| = |T_{n-1,j}| + |T_{n-1,j-1}|$

$T_{n,0}$

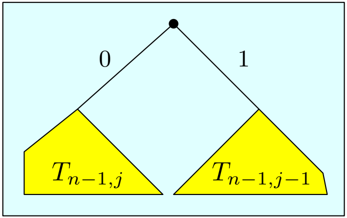


$T_{n,n}$



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

$T_{n,j}$

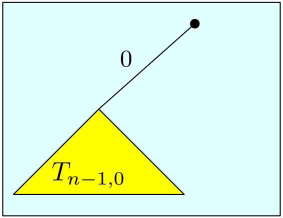


# of leafs:

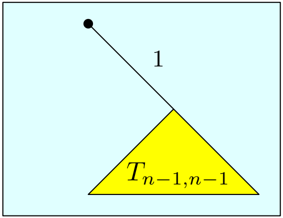
$$|T_{n,j}| = |T_{n-1,j}| + |T_{n-1,j-1}|$$

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

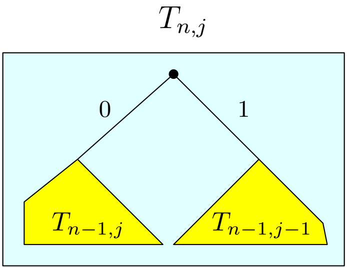
$T_{n,0}$



$T_{n,n}$



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

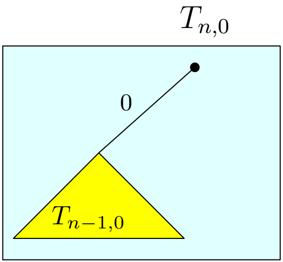


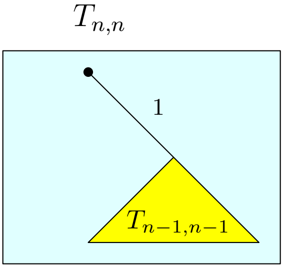
\# of leafs:

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

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

$$\implies |T_{n,j}| = \binom{n}{j}.$$





Encoding a string in $\mathbf{S_{n,j}}$

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 $S_{n,j}$.
5. **EncodeBinomCoeff**(s) denote this polytime procedure.

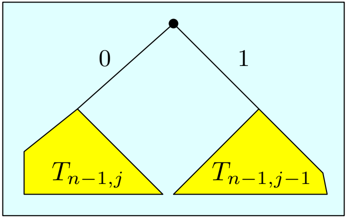
Encoding a string in $\mathbf{S_{n,j}}$

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 $S_{n,j}$.
5. **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. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.

$T_{n,j}$

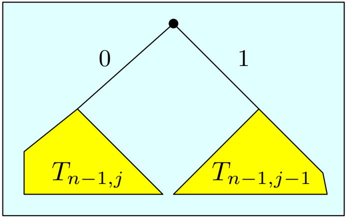


4. Input: $s \in S_{n,j}$: compute index of s in sorted set $S_{n,j}$.
5. **EncodeBinomCoeff**(s) denote this polytime procedure.

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. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.

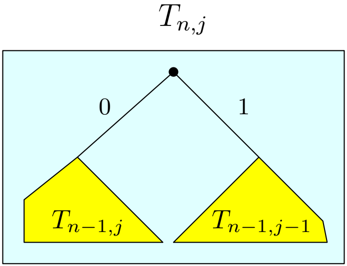
$T_{n,j}$



4. Input: $s \in S_{n,j}$: compute index of s in sorted set $S_{n,j}$.
5. **EncodeBinomCoeff**(s) denote this polytime procedure.

Encoding a string in $\mathbf{S_{n,j}}$

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 $S_{n,j}$.
5. **EncodeBinomCoeff**(s) denote this polytime procedure.

Decoding a string in $\mathbf{S_{n,j}}$

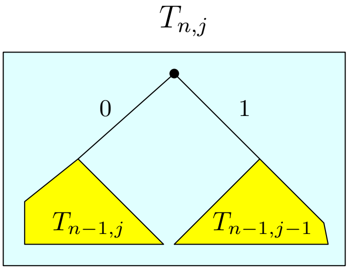
1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.
4. $x \in \left\{1, \ldots, \binom{n}{j}\right\}$: compute xth string in $S_{n,j}$ in polytime.
5. **DecodeBinomCoeff**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

Decoding a string in $\mathbf{S_{n,j}}$

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. $x \in \left\{1, \ldots, \binom{n}{j}\right\}$: compute xth string in $S_{n,j}$ in polytime.
5. **DecodeBinomCoeff** (x) denote this procedure.

Decoding a string in $\mathbf{S_{n,j}}$

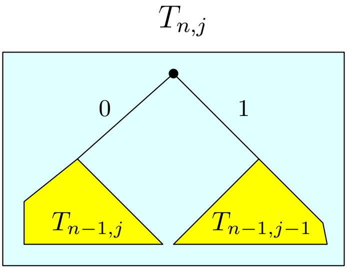
1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.



4. $x \in \left\{1, \ldots, \binom{n}{j}\right\}$: compute xth string in $S_{n,j}$ in polytime.
5. **DecodeBinomCoeff** (x) denote this procedure.

Decoding a string in $\mathbf{S_{n,j}}$

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.

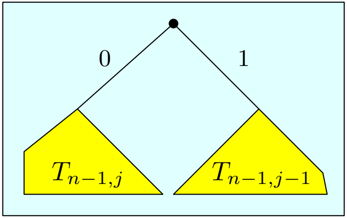


4. $x \in \left\{1, \ldots, \binom{n}{j}\right\}$: compute xth string in $S_{n,j}$ in polytime.
5. **DecodeBinomCoeff** (x) denote this procedure.

Decoding a string in $\mathbf{S_{n,j}}$

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

$T_{n,j}$



4. $\boldsymbol{x} \in \left\{ \mathbf{1}, \ldots, \binom{n}{j} \right\}$: compute $\boldsymbol{x}$th string in $\boldsymbol{S_{n,j}}$ in polytime.
5. **DecodeBinomCoeff** $(\boldsymbol{x})$ denote this procedure.

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**(α): *Input is string $\alpha \in S_{n,j}$, compute index x of α in $S_{n,j}$ in polynomial time in n.*
2. **DecodeBinomCoeff**(x): *Input index $x \in \left\{1, \ldots, \binom{n}{j}\right\}$. Output xth string α in $S_{n,j}$, in time $O(\operatorname{polylog} n + n)$.*

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.*

(B) *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 $S_j = \left\{1, \ldots, \binom{n}{j}\right\}$.
4. Given that input string s has j ones (out of n bits) defines a uniform distribution on $S_{n,j}$.
5. $x \leftarrow$ **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 $1, \ldots, N$.
9. Extract bits using **ExtractRandomness**(x, 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. $x \leftarrow$ **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 $1, \ldots, N$.
9. Extract bits using **ExtractRandomness**(x, 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. $x \leftarrow$ **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 $1, \ldots, N$.
9. Extract bits using **ExtractRandomness**(x, 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. $x \leftarrow$ **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 $1, \ldots, N$.
9. Extract bits using **ExtractRandomness**(x, 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. $x \leftarrow$ **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 $1, \ldots, N$.
9. Extract bits using **ExtractRandomness**(x, 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. $x \leftarrow$ **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 $1, \ldots, N$.
9. Extract bits using **ExtractRandomness**(x, 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. $x \leftarrow$ **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 $1, \ldots, N$.
9. Extract bits using **ExtractRandomness**(x, 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. $x \leftarrow$ **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 $1, \ldots, N$.
9. Extract bits using **ExtractRandomness**(x, 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. $x \leftarrow$ **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 $1, \ldots, N$.
9. Extract bits using **ExtractRandomness**(x, N):.

Exciting proof continued...

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

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

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

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

6. ... since $2^{n\mathbb{H}(p)}$ is a good approximation to $\binom{n}{np}$ as proved in previous lecture.

Exciting proof continued...

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

$$\mathrm{E}[B] = \sum_{k=0}^{n} \Pr[Z = k] \, \mathrm{E}\Big[B \;\Big|\; Z = k\Big],$$

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

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

6. ... since $2^{n\mathbb{H}(p)}$ is a good approximation to $\binom{n}{np}$ as proved in previous lecture.

Exciting proof continued...

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

$$\mathbf{E}\big[B\big] = \sum_{k=0}^{n} \Pr[Z = k]\, \mathbf{E}\Big[B \;\Big|\; Z = k\Big],$$

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

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

6. ... since $2^{n\mathbb{H}(p)}$ is a good approximation to $\binom{n}{np}$ as proved in previous lecture.

Exciting proof continued...

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

$$\mathbf{E}\left[B\right] = \sum_{k=0}^{n} \Pr[Z = k]\, \mathbf{E}\left[B \;\middle|\; Z = k\right],$$

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

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

6. ... since $2^{n\mathbb{H}(p)}$ is a good approximation to $\binom{n}{np}$ as proved in previous lecture.

Exciting proof continued...

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

$$\mathbf{E}\left[B\right] = \sum_{k=0}^{n} \Pr[Z = k]\, \mathbf{E}\left[B \;\middle|\; Z = k\right],$$

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

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

6. ... since $2^{n\mathbb{H}(p)}$ is a good approximation to $\binom{n}{np}$ as proved in previous lecture.

Exciting proof continued...

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

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

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

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

6. ... since $2^{n\mathbb{H}(p)}$ is a good approximation to $\binom{n}{np}$ as proved in previous lecture.

Exciting proof continued...

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

$$\mathbf{E}\big[B\big] = \sum_{k=0}^{n} \Pr[Z = k]\, \mathbf{E}\Big[B \;\Big|\; Z = k\Big],$$

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

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

6. ... since $2^{n\mathbb{H}(p)}$ is a good approximation to $\binom{n}{np}$ as proved in previous lecture.

Super exciting proof continued...

$$\mathbf{E}\left[B\right] = \sum_{k=0}^{n} \Pr[Z = k]\, \mathbf{E}\left[B \,\middle|\, Z = k\right].$$

$$\mathbf{E}\left[B\right] \geq \sum_{k=\lfloor n(p-\varepsilon)\rfloor}^{\lceil n(p+\varepsilon)\rceil} \Pr\left[Z = k\right] \mathbf{E}\left[B \,\middle|\, Z = k\right]$$

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

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

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

$$\geq \Big(n\mathbb{H}(p+\varepsilon) - \lg(n+1) - 2 \Big) \left(1 - 2\exp\left(-\frac{n\varepsilon^2}{4p} \right) \right),$$

since $\mu = \mathbf{E}[Z] = np$ and

$$\Pr\left[|Z - np| \geq \frac{\varepsilon}{p} pn \right] \leq 2\exp\left(-\frac{np}{4} \left(\frac{\varepsilon}{p} \right)^2 \right) = 2\exp\left(-\frac{n\varepsilon^2}{4p} \right),$$

Super exciting proof continued...

$$\mathbf{E}\left[B\right] = \sum_{k=0}^{n} \Pr[Z = k]\, \mathbf{E}\left[B \;\middle|\; Z = k\right].$$

$$\mathbf{E}\left[B\right] \geq \sum_{k=\lfloor n(p-\varepsilon)\rfloor}^{\lceil n(p+\varepsilon)\rceil} \Pr\left[Z = k\right] \mathbf{E}\left[B \;\middle|\; Z = k\right]$$

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

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

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

$$\geq \Big(n\mathbb{H}(p+\varepsilon) - \lg(n+1) - 2 \Big) \left(1 - 2\exp\left(-\frac{n\varepsilon^2}{4p} \right) \right),$$

since $\mu = \mathbf{E}[Z] = np$ and

$$\Pr\left[|Z - np| \geq \frac{\varepsilon}{p} pn \right] \leq 2\exp\left(-\frac{np}{4} \left(\frac{\varepsilon}{p} \right)^2 \right) = 2\exp\left(-\frac{n\varepsilon^2}{4p} \right),$$

Super exciting proof continued...

$$\mathbf{E}\big[B\big] = \sum_{k=0}^{n} \Pr[Z = k]\, \mathbf{E}\Big[B \,\Big|\, Z = k\Big].$$

$$\mathbf{E}\Big[B\Big] \geq \sum_{k=\lfloor n(p-\varepsilon)\rfloor}^{\lceil n(p+\varepsilon)\rceil} \Pr\Big[Z = k\Big]\, \mathbf{E}\Big[B \,\Big|\, Z = k\Big]$$

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

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

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

$$\geq \Big(n\mathbb{H}(p+\varepsilon) - \lg(n+1) - 2\Big)\left(1 - 2\exp\left(-\frac{n\varepsilon^2}{4p}\right)\right),$$

since $\mu = \mathbf{E}[Z] = np$ and

$$\Pr\left[|Z - np| \geq \tfrac{\varepsilon}{p} pn\right] \leq 2\exp\left(-\tfrac{np}{4}\left(\tfrac{\varepsilon}{p}\right)^2\right) = 2\exp\left(-\tfrac{n\varepsilon^2}{4p}\right),$$

Super exciting proof continued...

$$\mathbf{E}\Big[B\Big] = \sum_{k=0}^{n} \Pr[Z = k]\, \mathbf{E}\Big[B \,\Big|\, Z = k\Big].$$

$$\mathbf{E}\Big[B\Big] \geq \sum_{k=\lfloor n(p-\varepsilon)\rfloor}^{\lceil n(p+\varepsilon)\rceil} \Pr\Big[Z = k\Big]\, \mathbf{E}\Big[B \,\Big|\, Z = k\Big]$$

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

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

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

$$\geq \Big(n\mathbb{H}(p+\varepsilon) - \lg(n+1) - 2 \Big) \left(1 - 2\exp\left(-\frac{n\varepsilon^2}{4p} \right) \right),$$

since $\mu = \mathbf{E}[Z] = np$ and

$$\Pr\Big[|Z - np| \geq \tfrac{\varepsilon}{p} pn \Big] \leq 2\exp\left(-\frac{np}{4}\left(\frac{\varepsilon}{p}\right)^2 \right) = 2\exp\left(-\frac{n\varepsilon^2}{4p} \right),$$

Super exciting proof continued...

$$\mathbf{E}\left[B\right] = \sum_{k=0}^{n} \Pr[Z = k] \, \mathbf{E}\left[B \,\middle|\, Z = k\right].$$

$$\mathbf{E}\left[B\right] \geq \sum_{k=\lfloor n(p-\varepsilon)\rfloor}^{\lceil n(p+\varepsilon)\rceil} \Pr\left[Z = k\right] \mathbf{E}\left[B \,\middle|\, Z = k\right]$$

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

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

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

$$\geq \left(n\mathbb{H}(p+\varepsilon) - \lg(n+1) - 2 \right) \left(1 - 2\exp\left(-\frac{n\varepsilon^2}{4p} \right) \right),$$

since $\mu = \mathbf{E}[Z] = np$ and

$$\Pr\left[|Z - np| \geq \tfrac{\varepsilon}{p} pn \right] \leq 2\exp\left(-\tfrac{np}{4}\left(\tfrac{\varepsilon}{p}\right)^2 \right) = 2\exp\left(-\tfrac{n\varepsilon^2}{4p} \right),$$

Super exciting proof continued...

$$\mathbf{E}\Big[B\Big] = \sum_{k=0}^{n} \Pr[Z = k]\, \mathbf{E}\Big[B \;\Big|\; Z = k\Big].$$

$$\mathbf{E}\Big[B\Big] \geq \sum_{k=\lfloor n(p-\varepsilon)\rfloor}^{\lceil n(p+\varepsilon)\rceil} \Pr\Big[Z = k\Big]\, \mathbf{E}\Big[B \;\Big|\; Z = k\Big]$$

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

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

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

$$\geq \Big(n\mathbb{H}(p+\varepsilon) - \lg(n+1) - 2 \Big) \left(1 - 2\exp\left(-\frac{n\varepsilon^2}{4p} \right) \right),$$

since $\mu = \mathbf{E}[Z] = np$ and

$$\Pr\Big[|Z - np| \geq \tfrac{\varepsilon}{p} pn \Big] \leq 2\exp\left(-\frac{np}{4} \left(\frac{\varepsilon}{p} \right)^2 \right) = 2\exp\left(-\frac{n\varepsilon^2}{4p} \right),$$

Super exciting proof continued...

$$\mathbf{E}\left[B\right] = \sum_{k=0}^{n} \Pr[Z = k]\, \mathbf{E}\left[B \;\middle|\; Z = k\right].$$

$$\mathbf{E}\left[B\right] \geq \sum_{k=\lfloor n(p-\varepsilon)\rfloor}^{\lceil n(p+\varepsilon)\rceil} \Pr\left[Z = k\right] \mathbf{E}\left[B \;\middle|\; Z = k\right]$$

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

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

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

$$\geq \Bigl(n\mathbb{H}(p+\varepsilon) - \lg(n+1) - 2 \Bigr) \left(1 - 2\exp\left(-\frac{n\varepsilon^2}{4p} \right) \right),$$

since $\mu = \mathbf{E}[Z] = np$ and

$$\Pr\left[|Z - np| \geq \frac{\varepsilon}{p} pn \right] \leq 2\exp\left(-\frac{np}{4}\left(\frac{\varepsilon}{p}\right)^2 \right) = 2\exp\left(-\frac{n\varepsilon^2}{4p} \right),$$

Hyper super exciting proof continued...

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;

$$\mathbf{E}[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right) n\mathbb{H}(p)\left(1 - \frac{\delta}{10}\right)$$
$$\geq (1-\delta)n\mathbb{H}(p),$$

Hyper super exciting proof continued...

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;

$$\mathbf{E}[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right) n\mathbb{H}(p)\left(1 - \frac{\delta}{10}\right)$$
$$\geq (1-\delta)n\mathbb{H}(p),$$

Hyper super exciting proof continued...

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;

$$\mathbf{E}[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right) n\mathbb{H}(p) \left(1 - \frac{\delta}{10}\right)$$
$$\geq (1-\delta)n\mathbb{H}(p),$$

Hyper super exciting proof continued...

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;

$$\mathbf{E}[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right) n\mathbb{H}(p) \left(1 - \frac{\delta}{10}\right)$$
$$\geq (1-\delta) n\mathbb{H}(p),$$

Hyper super exciting proof continued...

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;

$$
\begin{aligned}
\mathbf{E}[B] &\geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right) n\mathbb{H}(p)\left(1 - \frac{\delta}{10}\right)\\
&\geq (1-\delta)n\mathbb{H}(p),
\end{aligned}
$$

Hyper super duper exciting proof continued...

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

Hyper super duper exciting proof continued...

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

Hyper super duper exciting proof continued...

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

Hyper super duper exciting proof continued...

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

Hyper super duper exciting proof continued...

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

Hyper super duper exciting proof continued...

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

Hyper super duper exciting proof continued...

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

Hyper super duper exciting proof continued...

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

Hyper super duper exciting proof continued...

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

■

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** $\texttt{Enc} : \{0,1\}^k \rightarrow \{0,1\}^n$ takes as input a sequence of k bits and outputs a sequence of n bits.
2. (k, n) **decoding function** $\texttt{Dec} : \{0,1\}^n \rightarrow \{0,1\}^k$ 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** $\texttt{Enc} : \{0,1\}^k \to \{0,1\}^n$ takes as input a sequence of k bits and outputs a sequence of n bits.
2. (k, n) **decoding function** $\texttt{Dec} : \{0,1\}^n \to \{0,1\}^k$ 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 γ for every possible k-bit input messages.

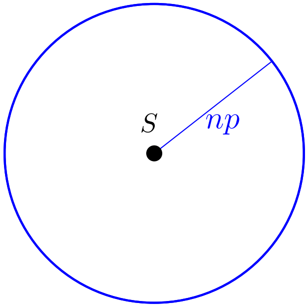
(ii) There are no (k, n) encoding and decoding functions with $k \geq n(1 - \mathbb{H}(p) + \delta)$ such that the probability of decoding correctly is at least γ for a k-bit input message chosen uniformly at random.

When the sender sends a string...

$S = s_1 s_2 \dots s_n$

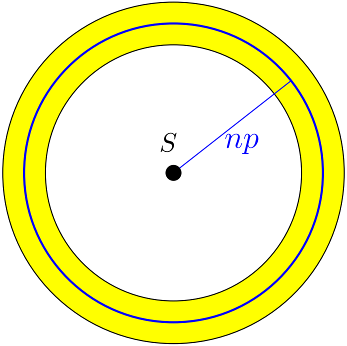
When the sender sends a string...

$S = s_1 s_2 \ldots s_n$



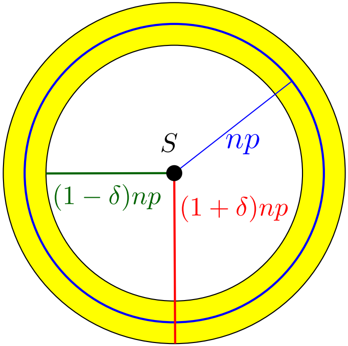
When the sender sends a string...

$\boldsymbol{S = s_1 s_2 \dots s_n}$



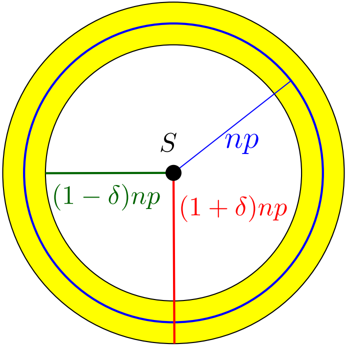
When the sender sends a string...

$S = s_1 s_2 \ldots s_n$



When the sender sends a string...

$S = s_1 s_2 \dots s_n$



One ring to rule them all!

Some intuition...

1. senders sent string $S = s_1 s_2 \dots s_n$.
2. receiver got string $T = t_1 t_2 \dots t_n$.
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: $\mathrm{E}[U] = pn$, and U is a binomial variable.
6. By Chernoff inequality: $U \in \left[(1-\delta)np, (1+\delta)np\right]$ with high probability, where δ 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

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2\binom{n}{(1+\delta)np} \leq \alpha = 2 \cdot 2^{n\mathbb{H}((1+\delta)p)}.$$

strings in it.

Some intuition...

1. senders sent string $S = s_1 s_2 \dots s_n$.
2. receiver got string $T = t_1 t_2 \dots t_n$.
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: $\mathbf{E}[U] = pn$, and U is a binomial variable.
6. By Chernoff inequality: $U \in \left[(1-\delta)np, (1+\delta)np\right]$ with high probability, where δ 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

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2\binom{n}{(1+\delta)np} \leq \alpha = 2 \cdot 2^{n\mathbb{H}((1+\delta)p)}.$$

strings in it.

Some intuition...

1. senders sent string $S = s_1 s_2 \dots s_n$.
2. receiver got string $T = t_1 t_2 \dots t_n$.
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: $U \in \left[(1-\delta)np, (1+\delta)np\right]$ with high probability, where δ 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

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2\binom{n}{(1+\delta)np} \leq \alpha = 2 \cdot 2^{n\mathbb{H}((1+\delta)p)}.$$

strings in it.

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: $U \in \left[(1-\delta)np, (1+\delta)np\right]$ with high probability, where δ 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

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2\binom{n}{(1+\delta)np} \leq \alpha = 2 \cdot 2^{n\mathbb{H}((1+\delta)p)}.$$

strings in it.

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: $\mathbf{E}[U] = pn$, and U is a binomial variable.
6. By Chernoff inequality: $U \in \left[(1-\delta)np, (1+\delta)np\right]$ with high probability, where δ 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
$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2\binom{n}{(1+\delta)np} \leq \alpha = 2 \cdot 2^{n\mathbb{H}((1+\delta)p)}.$$
strings in it.

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: $\mathbf{E}[U] = pn$, and U is a binomial variable.
6. By Chernoff inequality: $U \in \left[(1-\delta)np, (1+\delta)np\right]$ with high probability, where δ 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

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2\binom{n}{(1+\delta)np} \leq \alpha = 2 \cdot 2^{n\mathbb{H}((1+\delta)p)}.$$

strings in it.

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: $\mathbf{E}[U] = pn$, and U is a binomial variable.
6. By Chernoff inequality: $U \in \bigl[(1-\delta)np, (1+\delta)np\bigr]$ with high probability, where δ 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

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2\binom{n}{(1+\delta)np} \leq \alpha = 2 \cdot 2^{n\mathbb{H}((1+\delta)p)}.$$

strings in it.

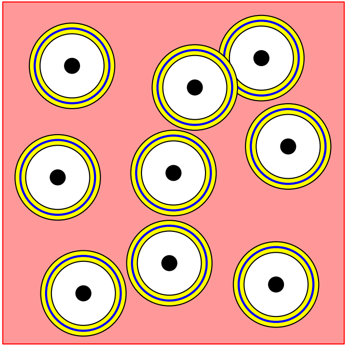
Some intuition...

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: $\mathbf{E}[U] = pn$, and U is a binomial variable.
6. By Chernoff inequality: $U \in \left[(1-\delta)np, (1+\delta)np\right]$ with high probability, where δ 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

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2\binom{n}{(1+\delta)np} \leq \alpha = 2 \cdot 2^{n\mathbb{H}((1+\delta)p)}.$$

strings in it.

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 $\implies \kappa \geq$

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

4. Consider all possible strings of length k such that $2^k \leq \kappa$.
5. Map ith string in $\{0,1\}^k$ to the center C_i of the ith 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?
$k = \lfloor \log \kappa \rfloor = n\Big(1 - \mathbb{H}\Big((1+\delta)p\Big)\Big) \approx n(1 - \mathbb{H}(p)),$

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 $\implies$ $\kappa \geq$

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

4. Consider all possible strings of length k such that $2^k \leq \kappa$.
5. Map ith string in $\{0,1\}^k$ to the center C_i of the ith 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?

$$k = \lfloor \log \kappa \rfloor = n\Big(1 - \mathbb{H}\Big((1+\delta)p\Big)\Big) \approx n(1 - \mathbb{H}(p)),$$

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 $\implies$ $\kappa \geq$
$$\kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{n\mathbb{H}((1+\delta)p)}} \approx 2^{n(1-\mathbb{H}((1+\delta)p))}.$$
4. Consider all possible strings of length k such that $2^k \leq \kappa$.
5. Map ith string in $\{0,1\}^k$ to the center C_i of the ith 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?
$$k = \lfloor \log \kappa \rfloor = n\Big(1 - \mathbb{H}\Big((1+\delta)p\Big)\Big) \approx n(1 - \mathbb{H}(p)),$$

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 $\implies \kappa \geq$

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

4. Consider all possible strings of length k such that $2^k \leq \kappa$.
5. Map ith string in $\{0,1\}^k$ to the center C_i of the ith 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?

$$k = \lfloor \log \kappa \rfloor = n\Big(1 - \mathbb{H}\Big((1+\delta)p\Big)\Big) \approx n(1 - \mathbb{H}(p)),$$

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 $\implies \kappa \geq$
$$\kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{n\mathbb{H}((1+\delta)p)}} \approx 2^{n(1-\mathbb{H}((1+\delta)p))}.$$
4. Consider all possible strings of length k such that $2^k \leq \kappa$.
5. Map ith string in $\{0,1\}^k$ to the center C_i of the ith 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?
$$k = \lfloor \log \kappa \rfloor = n\Big(1 - \mathbb{H}\Big((1+\delta)p\Big)\Big) \approx n(1 - \mathbb{H}(p)),$$

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 $\implies$ $\kappa \geq$
$$\kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{n\mathbb{H}((1+\delta)p)}} \approx 2^{n(1-\mathbb{H}((1+\delta)p))}.$$
4. Consider all possible strings of length k such that $2^k \leq \kappa$.
5. Map ith string in $\{0,1\}^k$ to the center C_i of the ith 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?
$$k = \lfloor \log \kappa \rfloor = n\Big(1 - \mathbb{H}\Big((1+\delta)p\Big)\Big) \approx n(1 - \mathbb{H}(p)),$$

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 $\implies \kappa \geq$
$$\kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{n\mathbb{H}((1+\delta)p)}} \approx 2^{n(1-\mathbb{H}((1+\delta)p))}.$$
4. Consider all possible strings of length k such that $2^k \leq \kappa$.
5. Map ith string in $\{0,1\}^k$ to the center C_i of the ith 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?
$$k = \lfloor \log \kappa \rfloor = n\Big(1 - \mathbb{H}\Big((1+\delta)p\Big)\Big) \approx n(1 - \mathbb{H}(p)),$$

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 $\implies \kappa \geq$
$$\kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{n\mathbb{H}((1+\delta)p)}} \approx 2^{n(1-\mathbb{H}((1+\delta)p))}.$$
4. Consider all possible strings of length k such that $2^k \leq \kappa$.
5. Map ith string in $\{0,1\}^k$ to the center C_i of the ith 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?
$$k = \lfloor \log \kappa \rfloor = n\Big(1 - \mathbb{H}\Big((1+\delta)p\Big)\Big) \approx n(1 - \mathbb{H}(p)),$$

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 $\implies$ $\kappa \geq$

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

4. Consider all possible strings of length k such that $2^k \leq \kappa$.
5. Map ith string in $\{0,1\}^k$ to the center C_i of the ith 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?

$$k = \lfloor \log \kappa \rfloor = n\Big(1 - \mathbb{H}\Big((1+\delta)p\Big)\Big) \approx n(1 - \mathbb{H}(p)),$$

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

Notes

Notes