We start by proving a generalization of Hamming Bound, which we did for the Hamming code towards the end of the last lecture.

**Lemma 1** (Hamming Bound). For an \((n, k, d)\) code \(C\),

\[
|C| \leq \frac{2^n}{\sum_{i=1}^{\lfloor \frac{d}{2} \rfloor} \binom{n}{i}}
\]

**Proof.** Let \(d\) be the minimum distance of the code \(C\). Consider the Hamming balls of radius \(\lfloor \frac{d-1}{2} \rfloor\) centred at the codewords. The number of strings in \(\{0, 1\}^n\) within this distance is at most \(\sum_{i=1}^{\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i}\). None of these balls intersect with each other, since otherwise the minimum distance will be less than \(d\). Thus each word is covered at most once (we may skip some). Thus if we count the "volume" of each of the Hamming balls:

\[
|C| \cdot \sum_{i=1}^{\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i} \leq 2^n
\]

The Hamming bound follows.

**Remark 2.** An intuitive interpretation of Hamming Bound is as a tradeoff result between \(n\) and \(k\). It shows the limits on on packing too many codewords in the space \(\{0, 1\}^n\) while keeping the minimum distance to be large. Thus, for a given minimum distance if we want to pack in too many codewords, we have no option but to increase the value of \(n\).

With this in mind, we turn our attention to Hamming codes. We considered the simple case in the last lecture. The parameters\(^1\) were \([7, 4, 3]\)

We know that the parity check matrix is given by:

\[
H = \begin{pmatrix}
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1
\end{pmatrix}
\]

\(^1\)We denote by \([n, k, d]_q\), the linear code mapping \(k\) bits to \(n\) bits, with minimum distance \(d\) and the field of size at most \(q\). \((n, k, d)\) denotes the same with no linearity restriction.
This indeed had the special structure that every column of three bit representations (except the all 0s) appears as columns. This observation is useful, as we will demonstrate below. Consider the hamming bound for this particular code.

\[ |C| \leq \frac{2^n}{n + 1} \]

We observe that, in a \([7, 4, 3]\) hamming code, \(|C| = 2^4\) and \(n = 7\). Thus the Hamming bound is met with an equality. This indeed, by our above remark indicates that Hamming codes pack in the optimal number of codewords into the space \(\{0, 1\}^n\) for the given distance. Is this a property possessed by only Hamming codes? Codes which meets the Hamming bound are called \textit{perfect codes.}

Thus, in our example set up, every word \(\{0, 1\}^n\) is within a Hamming ball of radius \(1 = \lceil \frac{d - 1}{2} \rceil\), there is a codeword. Thus, we can hope to correct one bit error in the received word. Since the error is at most one bit, we can received word as \(y = c + e_i\) where \(c\) is a code word and \(e_i\) is the \(i^{th}\) basis vector. Hence,

\[ Hy = H(c + e_i) = Hc + He_i = He_i = i \text{ in binary.} \]

Thus, the decoding for Hamming codes has an efficient algorithm even for locating the index for the error.

\textit{Remark 3.} How do we prove that the distance of the Hamming code is exactly 3? We use the characterization of minimum distance of a code as the minimum number of columns of the parity check matrix \((H)\) which are linearly dependent. Indeed the parity check matrix has three columns that are linearly dependent (take the first three columns). It also has the property that no two columns are linearly dependent (since they differ in at least one bit).

\section{Generalized Hamming Codes}

Now we generalize the construction. Let \(n\) be of the form \(2^r - 1\). We specify the code by describing the parity check matrix. The parity check matrix of the Hamming code is the matrix formed by all \(r\) bit representations of the numbers from 1, 2, \ldots 2\(^r\) – 1. Note that this excludes all 0s. Hence the parity check matrix is of dimensions \(2^r - 1 \times r\) which is \((n - k) \times k\). To calculate the distance, we can apply the same argument. The first three columns of the matrix are linearly dependent, and no two columns are linearly dependent. Hence the Hamming Code is an \([2^r - 1, 2^r - r - 1, 3]_2\) code.

We check the Hamming bound,

\[ |C| = 2^{2^r - 1 - r}. \]
We observe that any general hamming code of length $2^r - 1$ is also a perfect code.

Now we will see a close but complementary relative of the Hamming code, called the Hadamard codes. We introduce them through the concept of a Dual code.

## 2 Dual codes

Recall that the parity check matrix of a code generated by the generator matrix $G \in \{0,1\}^{n \times k}$, is defined as the matrix $H \in \{0,1\}^{(n-k) \times k}$. We have seen that any code can be uniquely identified by the generator matrix or the parity check matrix. $G : \mathbb{F}_2^k \to \mathbb{F}_2^n$

$C = \{y \in \mathbb{F}_2^n \wedge Hy = 0\}$

Now we consider the space that has its basis as the vectors in $H^\perp$. This space is the orthogonal space or dual space of $C$.

Given that the dimensionality of $C$ is $k$, it is directly observable that the dimensionality of the dual space is $n - k$. We also notice that if we consider the vectors in the dual matrix as a new code, the generator matrix of $C$ is the parity check matrix of the new code. We represent this new code by $C^\perp$.

Since $C^\perp$ is exactly the null space of $G$, we can also say that the set of all vectors that are linear combination of the vectors in $H$ are orthogonal to all vectors in $C$. In other words, $(C^\perp)^\perp = C$.

We can also have codes where $C = C^\perp$. Such codes are called **self-dual** codes.

In self-dual codes, each vector in the basis of the subspace is orthogonal to every other vector in the basis of the subspace, including itself.

## 3 Hadamard code

We now consider the dual code of Hamming code. $H^T$ is the generator matrix. If the message is $k$ bits in length, the rows of the generator matrix are all vectors $\in \{0,1\}^k$ except null vector. The code is the parity of all possible subsets of the $k$ bits, except $\phi$. The encoded string is of length $2^k - 1$.

Now consider the code with generator matrix as all vectors in $\{0,1\}^k$. This has the encoding as the parity of all subsets including $\phi$. This has the encoding of the Hamming code.

This code is called the **Hadamard code**.

Hadamard code, for message length $r$, has a block length of $2^r$. We observe that the rate of hadamard code $= \frac{r}{2^r}$, which is very bad in comparison to that of the Hamming code.
We now try to compute the minimum distance $d$ of the hadamard code. We use the following lemma for the same.

**Lemma 4.** If the weight of a codeword is defined as the number of non-zero symbols in it, the minimum distance $d$ of $C$ is exactly equal to the weight of the minimum weight non-zero codeword in $C$, where $c$ is a linear code.

**Proof.** The distance between any two codes is the number of bits in which they differ. In the given case, the distance between any two codes is the weight of the XOR of the two codes. Since the XOR of any two codes also lies in $C$, the weight of minimum weight codeword in $C \leq$ distance between any two codes in $C$.

Also, Weight of minimum weight codeword $w$ in $C = \delta(w, 0)$.

Hence, weight of minimum weight codeword = minimum distance.

**Distance of Hadamard code:** Now, let $x \neq 0$ be a $r$-bit message that has to be encoded using Hadamard code.

Consider $a$ to be any $r$-bit string. When we consider all possible $r$-bit strings for $a$, we consider all possible subsets of $x$ in the form of $x.a$.

Since $x$ is non-zero, there exists at least one bit where it is non-zero. Let $i$ be one such index such that $x_i \neq 0$

For $a$, consider the corresponding string that differs from $a$ only at the $i^{th}$ index. Let this string be $a'$.

Since $x_i = 1$ from our assumption, we notice that $x.a + x.a' = x.(a + a') = e_i$ where $e_i$ is a $r$-bit string with zero at every index except $i$. Hence, we can conclude that $x.a \neq x.a'$.

If we split the set of all $r$-bit strings into $2^{r-1}$ pairs, with each pair being an $a$ and corresponding $a'$ which differ in the fixed index $i$, we observe one of $x.a$ and $x.a'$ is equal to one. Since $x.a$ and $x.a'$ are the parities of some two subsets of $x$, they are two bits in the encoding. This means that exactly half the number of bits of the encoding of $x$ will be 1.

In other words, weight of each encoding = $2^{r-1}$.

From the lemma, minimum distance associated with Hadamard code = $2^{r-1}$, which is equal to half the length of the encoding itself.

Relative distance of Hadamard code = $\frac{n}{2}$.

**4 Singleton bound**

**Lemma 5.** For any code $(n,k,d)$, the distance $d \leq n - k + 1$.

**Proof.** Let $E$ be the encoding of the code.

$E : \{0,1\}^k \rightarrow \{0,1\}^n$. 

B04-4
We consider a projection $\pi$ of $k - 1$ bits of the encoded string. By projection, we mean we consider only the first $k - 1$ bits and ignore the rest.

Since the original message is $k$ bits in length, by pigeon hole principle, there exists atleast 2 codes $C_1$ and $C_2$, such that, $\pi(E(C_1)) = \pi(E(C_2))$.

These two codes can differ in maximum of $n - (k - 1)$ bits.

$\Rightarrow d \leq n - k + 1.$

\section{Reed-Solomon codes}

We now change our focus from binary codes to higher order codes.

We define the encoding as follows. Consider set $S = \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \subseteq F$. If the message symbols are $m_0, m_1, \ldots, m_{k-1}$, and $p(x)$ is some polynomial given by $p(x) = m_0x^0 + m_1x^1 + m_2x^2 + \cdots + m_{k-1}x^{k-1}$.

Encoding ($E$) is given by $E(x) = (p(\alpha_1), p(\alpha_2), \ldots, p(\alpha_n))$.

We want maximum $d$. In other words, we want a large number of non-zeroes in the code. Since $\alpha_i$ are unique and $p$ is a $k - 1$ degree polynomial in one variable, there can be only $k - 1$ zeroes.

Given two codes $x$ and $y$, since $S$ is a fixed set, we can conclude that the encodings are equal at only the zeroes. The two codes can have the same value in only atmost $k - 1$ places.

The distance $d \geq (n - k + 1)$.

$\Rightarrow$ Reed-Solomon codes satisfy equality in the singleton bound.

Also, we can write the generator matrix of this code as

\[ G = \begin{pmatrix}
\alpha_0^0 & \alpha_1^0 & \cdots & \alpha_1^{k-1} \\
\alpha_0^1 & \alpha_1^1 & \cdots & \alpha_2^{k-1} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha_0^{n-1} & \alpha_1^{n-1} & \cdots & \alpha_n^{k-1}
\end{pmatrix} \]

Hence, we can conclude that the code is a linear code.

For maximum distance, we maximize the value of $n - k$. But this affects the rate of the code as well. We have to choose the value of $n$ such that there is optimum rate as well as a reasonably high distance.

We try to reduce code from $\mathcal{F}^k$ to $\mathcal{F}^2$. We can represent each element in binary, but the code may not be linear anymore. It can be linear if the field size is a power of 2. We still haven’t achieved singleton bound in binary.