

---

**16.36: Ingeniería de sistemas de comunicación**

**Clases 12/13: Codificación y capacidad de canal**

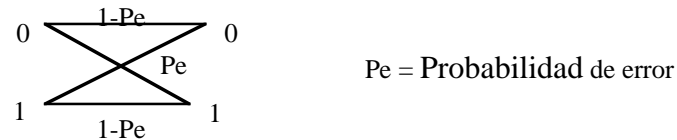
**Eytan Modiano**

# Codificación de canal

---

- Al transmitir por un canal con ruido, algunos bits se reciben con errores

## Ejemplo : Canal Simétrico Binario (BSC)



- P: ¿Cómo se pueden eliminar estos errores?
- R: Codificación: la suma de bits redundantes que nos ayuda a determinar con gran precisión lo que se envió

# Ejemplo (Código de repetición)

---

Repita cada bit n veces (n-impar)

Entrada	Codificación
0	000.....0
1	11.....1

Decodificador:

- Si la secuencia recibida contiene  $n/2$  o más 1 decodifique como un 1 y 0
  - Decodificación de Verificación Máxima

$$\begin{aligned} P(\text{error} \mid 1 \text{ enviado}) &= P(\text{error} \mid 0 \text{ enviado}) \\ &= P[\text{ocurren más de } n/2 \text{ errores de bit}] \end{aligned}$$

$$= \sum_{i=\lceil n/2 \rceil}^n \binom{n}{i} P_e^i (1 - P_e)^{n-i}$$

# Código de repetición, (cont.)

---

- Para  $P_e < 1/2$ ,  $P(\text{error})$  es decreciente en  $n$ 
  - $\Rightarrow$  para cualquier  $\varepsilon$ ,  $\exists n$  suficientemente grande para que  $P(\text{error}) < \varepsilon$

## Tasa de código: relación entre bits de datos y bits transmitidos

- Para el código de repetición  $R = 1/n$
- Para enviar un bit de datos se han de transmitir  $n$  bits de canal “expansión de banda ancha”
- En general, un código  $(n,k)$  utiliza  $n$  bits de canal para transmitir  $k$  bits de datos
  - Tasa de código  $R = k / n$
- Objetivo: para una probabilidad de error deseada,  $\varepsilon$ , hallar la mayor tasa de código que puede lograr  $p(\text{error}) < \varepsilon$

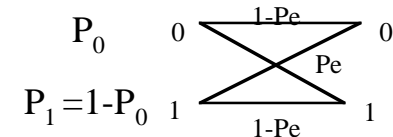
# Capacidad de canal

- La capacidad de un canal discreto sin memoria viene dada por:

$$C = \max_{p(x)} I(X;Y)$$



## Ejemplo : Canal Simétrico Binario (BSC)



$$I(X;Y) = H(Y) - H(Y|X) = H(X) - H(X|Y)$$

$$H(X|Y) = H(X|Y=0) \cdot P(Y=0) + H(X|Y=1) \cdot P(Y=1)$$

$$H(X|Y=0) = H(X|Y=1) = P_e \log(1/P_e) + (1-P_e) \log(1/1-P_e) = H_b(P_e)$$

$$H(X|Y) = H_b(P_e) \Rightarrow I(X;Y) = H(X) - H_b(P_e)$$

$$H(X) = P_0 \log(1/P_0) + (1-P_0) \log(1/1-P_0) = H_b(p_0)$$

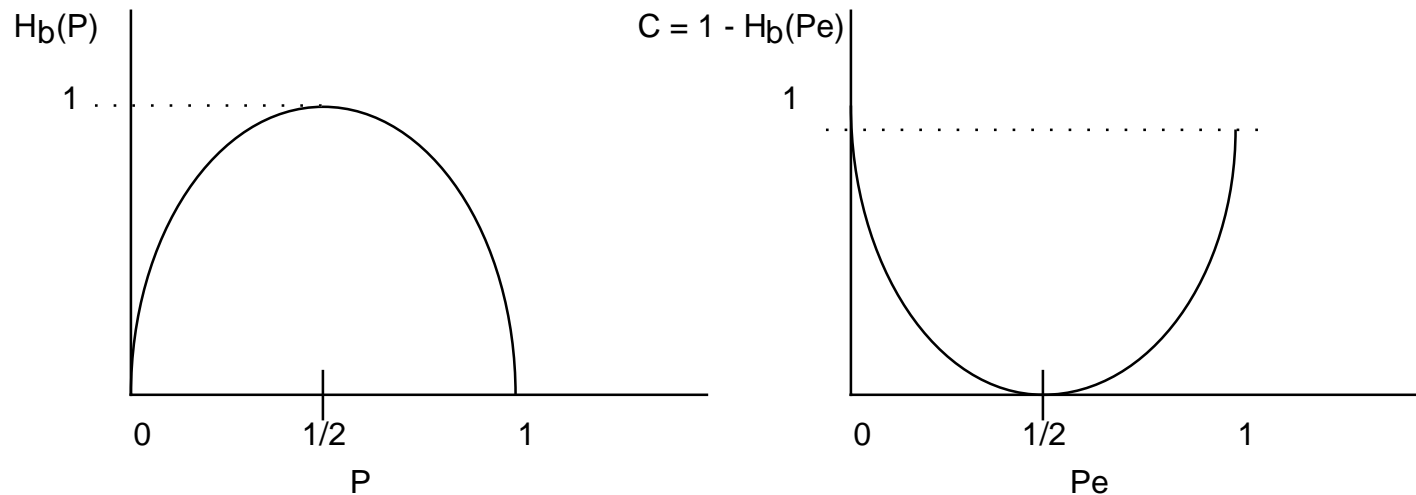
$$\Rightarrow I(X;Y) = H_b(P_0) - H_b(P_e)$$

# Capacidad del BSC

$$I(X;Y) = H_b(P_0) - H_b(P_e)$$

- $H_b(P) = P \log(1/P) + (1-P) \log(1/1-P)$ 
  - $H_b(P) \leq 1$  con igualdad si  $P=1/2$

$$C = \max_{P_0} \{I(X;Y) = H_b(P_0) - H_b(P_e)\} = 1 - H_b(P_e)$$



**$C = 0$  cuando  $P_e = 1/2$  y  $C = 1$  cuando  $P_e = 0$  o  $P_e=1$**

# Teorema de codificación de canal (Claude Shannon)

---

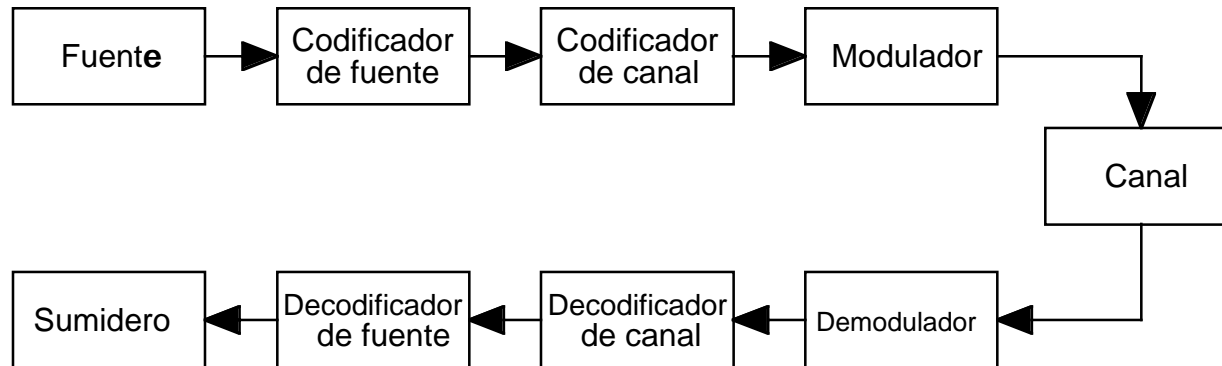
**Teorema:** para todo  $R < C$  y  $\varepsilon > 0$  existe un código de tasa  $R$  cuya probabilidad de error es  $< \varepsilon$

- $\varepsilon$  puede ser arbitrariamente pequeño
- La demostración utiliza una  $n$  de bloque de gran tamaño  
a medida que  $n \rightarrow \infty$  se logra la capacidad
- **En la práctica, los códigos que logran capacidad son difíciles de hallar**
  - El objetivo es hallar un código que se acerque lo más posible a lograr la capacidad
- **Inversa del teorema de codificación:**
  - Para todos los códigos de tasa  $R > C$ ,  $\exists \varepsilon_0 > 0$ , tal que la probabilidad de error es siempre mayor que  $\varepsilon_0$   
Para tasas de código mayores que la capacidad, la probabilidad de error queda delimitada más allá de 0

# Codificación de canal

---

- Diagrama de bloque



# Enfoques de la codificación

---

- **Códigos de bloque**
  - Los datos se dividen en bloques de la misma longitud
  - Cada bloque se “mapea” con otro bloque mayor

**Ejemplo: (6,3) codificación,  $n = 6$ ,  $k = 3$ ,  $R = 1/2$**

**000 → 000000**

**100 → 100101**

**001 → 001011**

**101 → 101110**

**010 → 010111**

**110 → 110010**

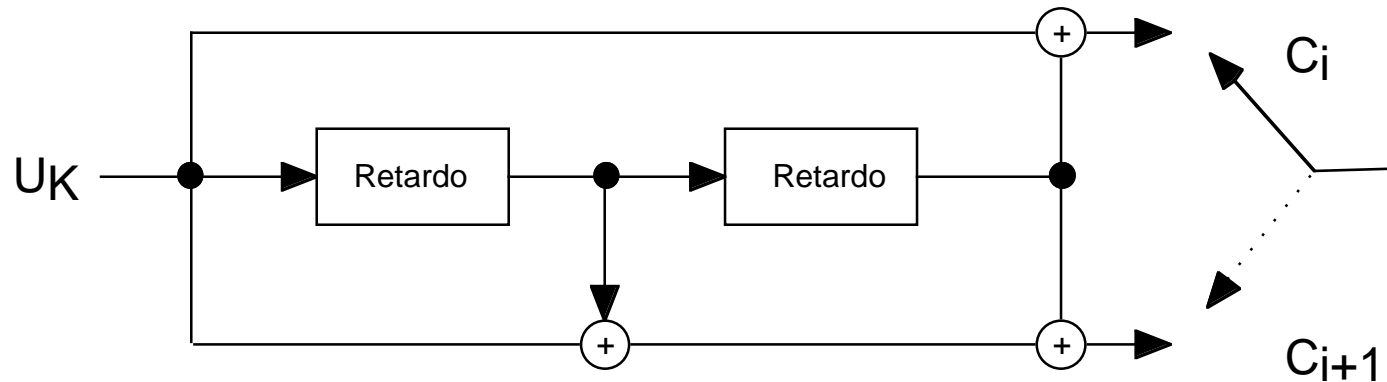
**011 → 011100**

**111 → 111001**

- **Un (n,k) código de bloque binario es una colección de  $2^k$  n-tuples binarios ( $n > k$ )**
  - $n$  = longitud del bloque
  - $k$  = número de bits de datos
  - $n-k$  = número de bits comprobados
  - $R = k / n$  = tasa de código

# Enfoques de la codificación

- **Códigos convolucionales**
  - La salida se facilita observando una ventana deslizante de entrada



$$C_{(2K)} = U_{(2K)} \oplus U_{(2K-2)}, \quad C_{(2K+1)} = U_{(2K+1)} \oplus U_{(2K)} \oplus U_{(2K-1)}$$

$$\oplus \text{ mod}(2) \text{ suma } (1+1=0)$$

# Códigos de bloque

---

- Un código de bloque es sistemático si las palabras de código se pueden dividir en una parte de datos y en una parte redundante
  - El código anterior (6,3) era sistemático

## Definiciones:

- Dado  $X \in \{0,1\}^n$ , el **Peso de Hamming** de  $X$  es el número de unos en  $X$
- Dado  $X, Y \in \{0,1\}^n$ , la **Distancia de Hamming** entre  $X$  e  $Y$  es el número de posiciones en los que se diferencian,

$$d_H(X, Y) = \sum_{i=1}^n X_i \oplus Y_i = \text{Weight}(X + Y)$$

$$X + Y = [x_1 \oplus y_1, x_2 \oplus y_2, \dots, x_n \oplus y_n]$$

- La **distancia mínima** de un código es la Distancia Hamming entre las dos palabras código más cercanas:

$$d_{\min} = \min \{d_H(C_1, C_2)\} \\ C_1, C_2 \in C$$

# Decodificación

---



- Es posible que  $r$  no sea igual a  $u$  debido a errores de transmisión
- Dado  $r$ , ¿cómo sabemos qué palabra código se envió?

**Decodificación por máxima verosimilitud:**

Mapear el  $n$ -tuple recibido  $r$  con la palabra código  $C$  que maximice,  
 $P \{ r \mid C \text{ se transmitió} \}$

**Decodificación por distancia mínima (vecino más próximo)**

Mapear  $r$  con la palabra código  $C$  de modo tal que se minimice la distancia hamming entre  $r$  y  $C$  (p. ej.,  $\min d_H(r, C)$ )

⇒ Para la mayoría de los canales, la decodificación por distancia mínima es la misma que la decodificación por máxima verosimilitud

# Códigos de bloque lineales

---

- Un  $(n,k)$  código de bloque lineal (LBC) se define por  $2^k$  palabras de código de longitud  $n$

$$C = \{ C_1, \dots, C_m \}$$

- Un  $(n,k)$  LBC es un subespacio  $K$ -dimensional de  $\{0,1\}^n$ 
  - $(0 \dots 0)$  siempre es una palabra código
  - If  $C_1, C_2 \in C$ ,  $C_1 + C_2 \in C$

- Teorema: para un LBC la distancia mínima es igual al peso mínimo ( $W_{\min}$ ) del código

$$W_{\min} = \min_{(\text{sobre todo } C_i)} \text{Peso}(C_i)$$

Demostración: suponga que  $d_{\min} = d_H(C_i, C_j)$ , donde  $C_1, C_2 \in C$

$$d_H(C_i, C_j) = \text{Peso}(C_i + C_j),$$

pero como  $C$  es un LBC entonces  $C_i + C_j$  es también una palabra código

# Códigos sistemáticos

---

**Teorema:** cualquier  $(n,k)$  LBC se puede representar de forma sistemática  
si:  $\text{datos} = x_1 \dots x_k$ ,  $\text{palabra código} = x_1 \dots x_k c_{k+1} \dots x_n$

- Por lo tanto hablaremos sólo de los códigos sistemáticos
- Las palabras código correspondientes a las secuencias de información:  
 $e_1 = (1,0,\dots,0)$ ,  $e_2 = (0,1,0,\dots,0)$ ,  $e_k = (0,0,\dots,1)$  como base del código
  - Sin duda, son independientes linealmente
  - Los  $n$ -tuples  $K$  linealmente independientes definen completamente el subespacio dimensional  $K$  que forma el código

**Secuencia de información**

$$e_1 = (1,0,\dots,0)$$

$$e_2 = (0,1,0,\dots,0)$$

$$e_k = (0,0,\dots,1)$$

**Palabra de código**

$$g_1 = (1,0,\dots,0, g_{(1,k+1)} \dots g_{(1,n)})$$

$$g_2 = (0,1,\dots,0, g_{(2,k+1)} \dots g_{(2,n)})$$

$$g_k = (0,0,\dots,0, g_{(k,k+1)} \dots g_{(k,n)})$$

- $g_1, g_2, \dots, g_k$  forma una base para el código

# La matriz generadora

---

$$G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & & & g_{2n} \\ \vdots & & & \\ g_{k1} & & & g_{kn} \end{bmatrix}$$

- Para la secuencia de entrada  $x = (x_1, \dots, x_k)$ :  $C_x = xG$ 
  - Cada palabra código es una combinación lineal de las filas de  $G$
  - La palabra de código que corresponde a cada secuencia de entrada se puede derivar de  $G$
  - Como cada entrada se puede representar como una combinación lineal de la base  $(e_1, e_2, \dots, e_k)$ , las palabras de código correspondientes se pueden representar como una combinación lineal de las filas correspondientes de  $G$
- Nota:  $x_1 \leftrightarrow C_1, x_2 \leftrightarrow C_2 \Rightarrow x_1 + x_2 \leftrightarrow C_1 + C_2$

# Ejemplo

---

- Considere el código (6,3) anterior:

100 → 100101;      010 → 010111;      001 → 001011

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Palabra de código para (1,0,1) = (1,0,1)G = (1,0,1,1,1,0)

$$G = \left[ \begin{array}{c|c} I_K & P_{K \times (n-K)} \end{array} \right]$$

$I_K = K \times K$  matriz de identidad

# Matriz de chequeo de paridad

---

$$H = \left[ \begin{array}{c|c} P^T & I_{(n-K)} \end{array} \right]$$

$I_{(n-K)}$  =  $(n - K) \times (n - K)$  matriz de identidad

Ejemplo:

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Ahora, si  $c_i$  es una palabra código de  $C$ , entonces,  $c_i H^T = \vec{0}$

- “ $C$  está en el espacio nulo de  $H$ ”
- Cualquier palabra código en  $C$  es ortogonal a las filas de  $H$

# Decodificación

---

- $v$  = palabra de código transmitida =  $v_1 \dots v_n$
- $r$  = palabra de código recibida =  $r_1 \dots r_n$
- $e$  = patrón de error =  $e_1 \dots e_n$
- $r = v + e$
- $S = rH^T$  = Síndrome de  $r$   
 $= (v+e)H^T = vH^T + eH^T = eH^T$
- $S$  es igual a '0' si y sólo si  $e \in C$ 
  - P. ej., el patrón de error es una palabra código
- $S \neq 0 \Rightarrow$  error detectado
- $S = 0 \Rightarrow$  no se detectan errores (es posible que hayan ocurrido y no se hayan detectado)
- Suponga que  $S \neq 0$ , ¿cómo podemos saber cuál fue verdaderamente la palabra código transmitida?

# Decodificación basada en el síndrome

---

- **Es posible que muchos patrones de error hayan creado el mismo síndrome**  
Para el patrón de error  $e_0 \Rightarrow S_0 = e_0 H^T$   
  
Considere el patrón de error  $e_0 + c_i$  ( $c_i \in \mathbb{C}$ )  
$$S'_0 = (e_0 + c_i)H^T = e_0 H^T + c_i H^T = e_0 H^T = S_0$$
- **Así, para un patrón de error,  $e_0$ , el resto de los patrones de error que pueden expresarse como  $e_0 + c_i$  para cierto  $c_i \in \mathbb{C}$  son también patrones de error con el mismo síndrome**
- **Para un síndrome dado, no podemos decir qué patrón de error ocurrió en realidad, pero el más probable es el de peso mínimo**
  - Decodificación por distancia mínima
- **Para un síndrome dado, hallar el patrón de error de peso mínimo ( $e_{\min}$ ) que da este síndrome y decodificar:  $r' = r + e_{\min}$**

# Formación habitual

---

$M = 2^K$	$C_1$	$C_2$	$\dots$	$C_M$	<i>Síndrome</i>
	$e_1$	$e_1 + C_2$		$e_1 + C_M$	$S_1$
	$\vdots$	$e_2 + C_2$		$e_2 + C_M$	$S_2$
	$e_{2^{(n-K)}-1}$				$S_{2^{(n-K)}-1}$

- La fila 1 está compuesta por todas las palabras código M
- La fila 2  $e_1 = n$ -tuple de peso mínimo no presente en la formación
  - P. ej., el patrón de error de peso mínimo
- La fila  $i$ ,  $e_i = n$ -tuple de peso mínimo no presente en la formación
- Todos los elementos de cualquier fila tienen el mismo síndrome
  - Los elementos de una fila se llaman “co-sets”
- El primer elemento de cada fila es el patrón de error de peso mínimo con dicho síndrome
  - Denominado “co-set principal”

# Algoritmo de descodificación

---

- Recibir vector  $r$ 
  - 1) Hallar  $S = rH^T =$  síndrome de  $r$
  - 2) Hallar el co-set principal  $e$ , correspondiente a  $S$
  - 3) Decodificar:  $C = r+e$
- “Decodificación por distancia mínima”
  - Decodificar en la palabra de código más próxima a la secuencia recibida

# Ejemplo (decodificación de síndrome)

- Código sencillo (4,2)

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Datos      palabra código

00            0000  
 01            0101  
 10            1010  
 11            1111

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad H^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

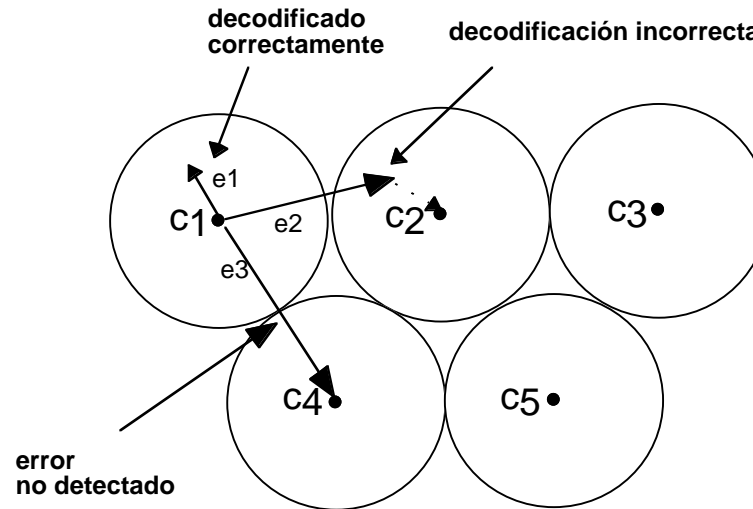
Formación habitual	0000	0101	1010	1111	Síndrome
	1000	1101	0010	0111	10
	0100	0001	1110	1011	01
	1100	1001	0110	0011	11

Suponga que se recibe 0111, S = 10, co-set principal = 1000

Decodificar: C = 0111 + 1000 = 1111

# Decodificación por distancia mínima

---



- La decodificación por distancia mínima mapea una secuencia recibida con la palabra de código más próxima
- Si un patrón de error mapea la palabra de código enviada con otra palabra de código válida, dicho error no será detectado (p.ej., e3)
  - Cualquier patrón de error que equivalga a una palabra de código dará errores no detectados
- Si un patrón de error mapea la secuencia enviada con la esfera de otra palabra de código, será decodificada incorrectamente (p.ej., e2)

# Rendimiento de códigos de bloque

---

- **Detección de errores:** calcule el síndrome,  $S \neq 0 \Rightarrow$  error detectado
  - Solicitar retransmisión
  - Utilizado en redes de paquetes
- **Un código de bloque lineal detectará todos los patrones de error que no sean palabras de código**
- **Corrección de errores:** decodificación basada en el síndrome
  - Todos los patrones de error de peso  $< d_{\min}/2$  se decodificarán correctamente
  - Por eso es importante diseñar códigos con una distancia mínima grande ( $d_{\min}$ )
  - Cuanto más grande sea la distancia mínima más pequeña es la probabilidad de decodificación incorrecta

# Códigos de Hamming

---

- **Código de bloque lineal capaz de corregir errores únicos**
  - $n = 2^m - 1$ ,  $k = 2^m - 1 - m$   
(e.g., (3,1), (7,4), (15,11)...)
  - $R = 1 - m/(2^m - 1) \Rightarrow$  frecuencia muy alta
  - $d_{\min} = 3 \Rightarrow$  corrección de error único
- **Construcción de códigos de Hamming**
  - La matriz de chequeo de paridad (H) se compone de todos los m-tuples binarios distintos de cero

**Ejemplo: (7,4) código de Hamming (m=3)**

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$