

# Colisiones

Dado que  $A \neq B$ ,

Dijimos que una función de hash tenía una **colisión** cuando

$$h(A) = h(B)$$

# Colisiones



Eso significa que  $A$  y  $B$  caen en la misma celda de la tabla

¿Cómo podemos guardar ambos datos en la tabla?

Nos interesa poder buscarlos en el futuro

# Direccionamiento Abierto



Podemos buscar otra celda que esté disponible

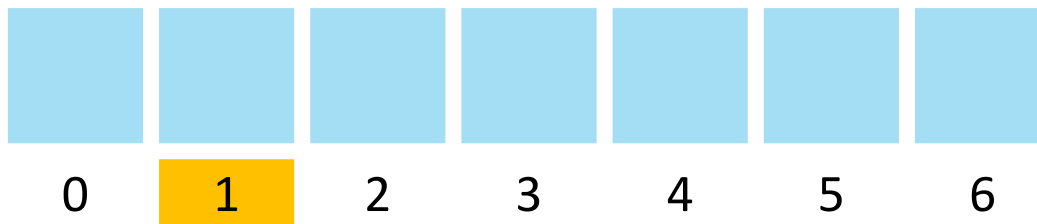
¿Dónde la buscamos? Debemos seguir alguna regla

¿Qué complejidad tiene esto?

# Sondeo Lineal

$$m = 7$$

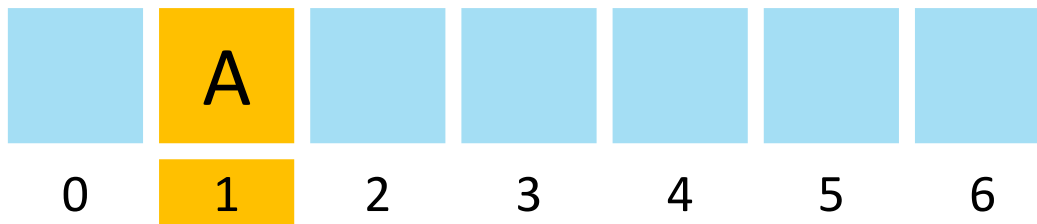
Insertemos la A.  $h(A) = 15; 15 \bmod 7 = 1$



# Sondeo Lineal

$$m = 7$$

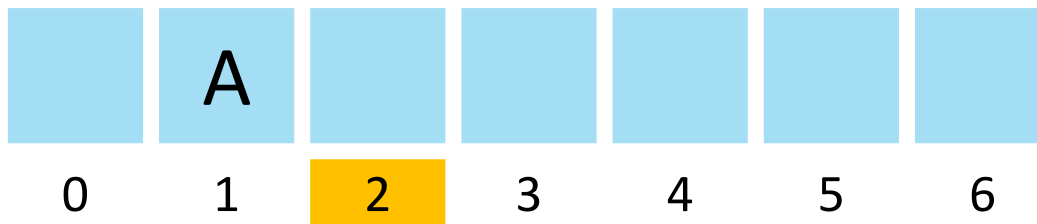
Insertemos la A.  $h(A) = 15; 15 \bmod 7 = 1$



# Sondeo Lineal

$$m = 7$$

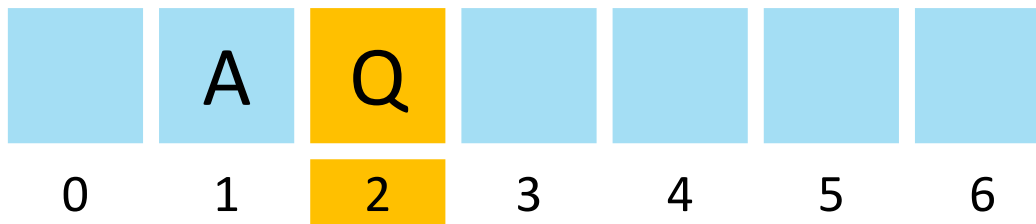
Insertemos la Q.  $h(Q) = 37; 37 \bmod 7 = 2$



# Sondeo Lineal

$$m = 7$$

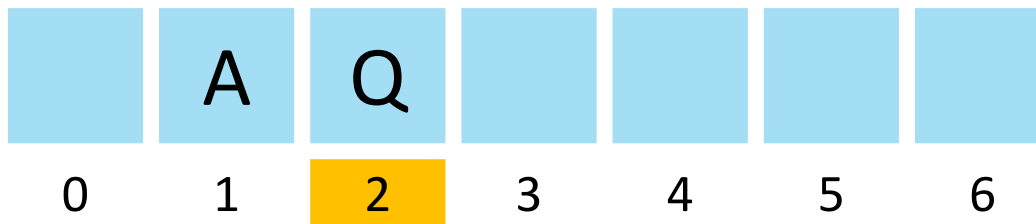
Insertemos la Q.  $h(Q) = 37; 37 \bmod 7 = 2$



# Sondeo Lineal

$$m = 7$$

Insertemos la L.  $h(L) = 51$ ;  $51 \bmod 7 = 2$

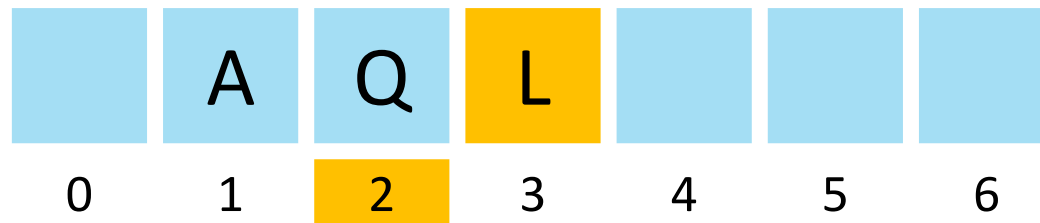




# Sondeo Lineal

$$m = 7$$

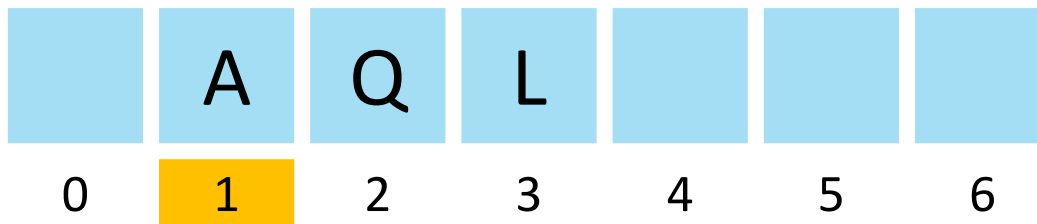
Insertemos la L.  $h(L) = 51$ ;  $51 \bmod 7 = 2$



# Sondeo Lineal

$$m = 7$$

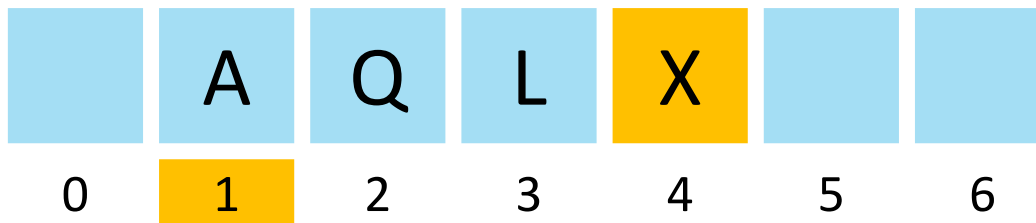
Insertemos la X.  $h(X) = 29$ ;  $29 \bmod 7 = 1$



# Sondeo Lineal

$$m = 7$$

Insertemos la X.  $h(X) = 29$ ;  $29 \bmod 7 = 1$



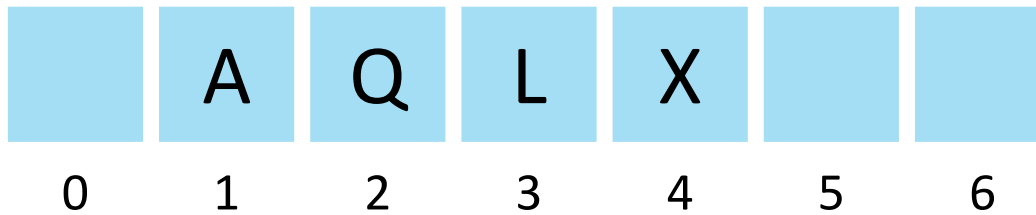
# Direcccionamiento Abierto

Métodos populares de direccionamiento abierto son:

- Sondeo Lineal
  - Buscar en  $H$ ,  $H + 1$ ,  $H + 2$ ,  $H + 3$  ...
- Sondeo Cuadrático:
  - Buscar en  $H$ ,  $H + 1c_1 + 1^2c_2$ ,  $H + 2c_1 + 2^2c_2$  ...
- *Double hashing*:
  - Buscar en  $h_1(k)$ ,  $h_1(k) + h_2(k)$ ,  $h_1(k) + 2h_2(k)$ , ...

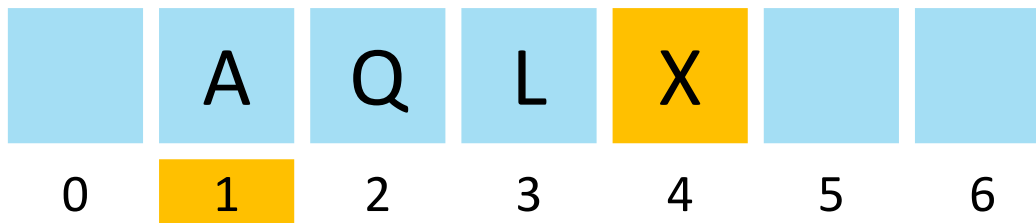
# Búsqueda

$$m = 7$$



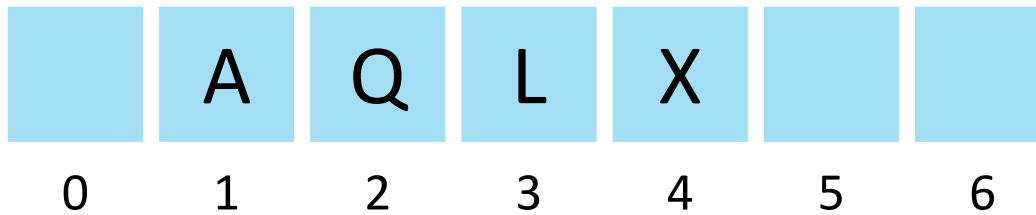
¿Cómo buscamos la X con sondeo lineal?

$$h(X) = 29; 29 \bmod 7 = 1$$



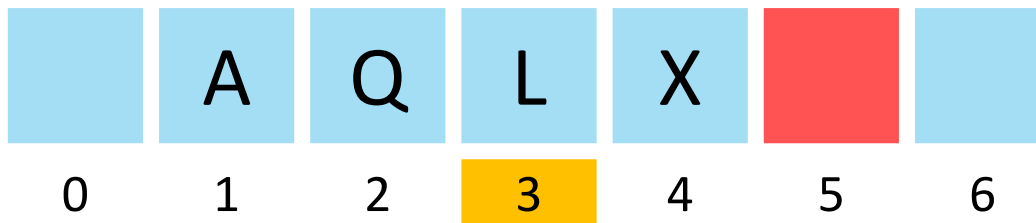
# Búsqueda

$$m = 7$$



¿Cómo buscamos la R con sondeo lineal?

$$h(R) = 10; 10 \bmod 7 = 3$$



# Direccionamiento Abierto



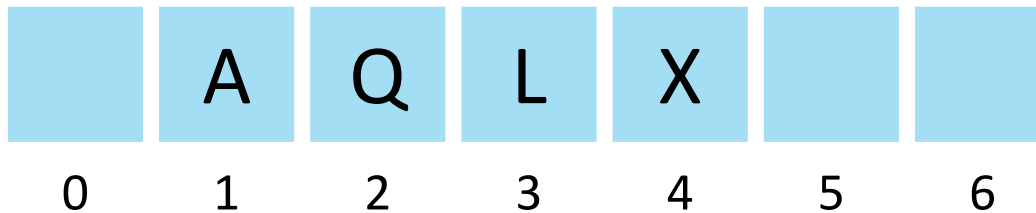
¿Qué problema tiene el sondeo lineal?

¿Qué problema tienen los otros esquemas?

¿Qué problema tiene el guardar los datos en otra celda?

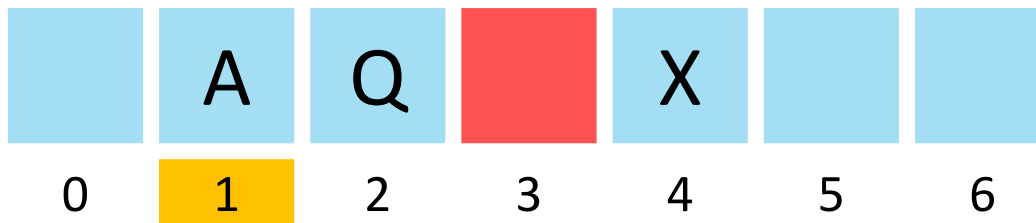
# Eliminación

$$m = 7$$



Eliminemos la L. ¿Cómo buscamos la X con sondeo lineal?

$$h(X) = 29; 29 \bmod 7 = 1$$





# Direccionamiento cerrado



Si  $A$  ya estaba en la tabla:

Podemos guardar  $A$  y  $B$  en la misma celda...

¡Dentro de otra estructura de datos!

# Encadenamiento



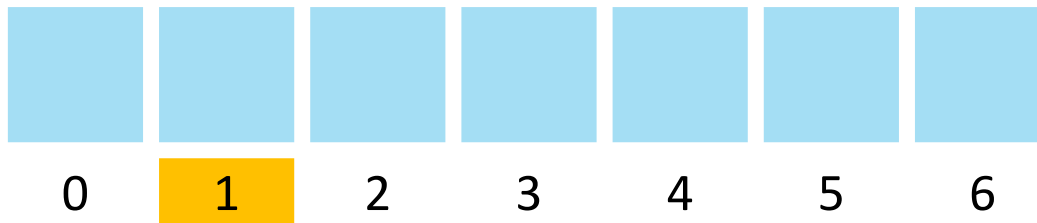
Si tenemos una **lista** en cada celda de la tabla

Hemos guardado  $n$  datos, y la tabla es de tamaño  $m$

¿Cuál sería la complejidad de las operaciones de la tabla?

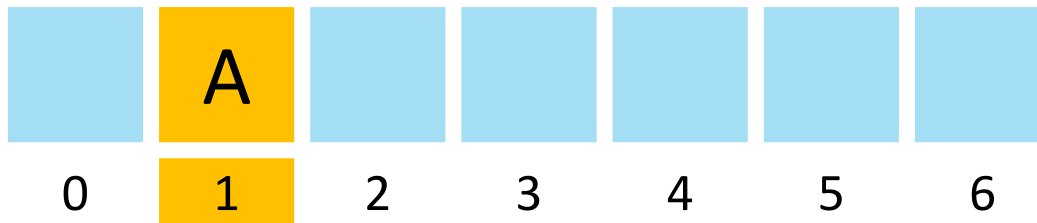
# Encadenamiento

$m = 7$ . Insertemos la A.  $h(A) = 15$ ;  $15 \bmod 7 = 1$



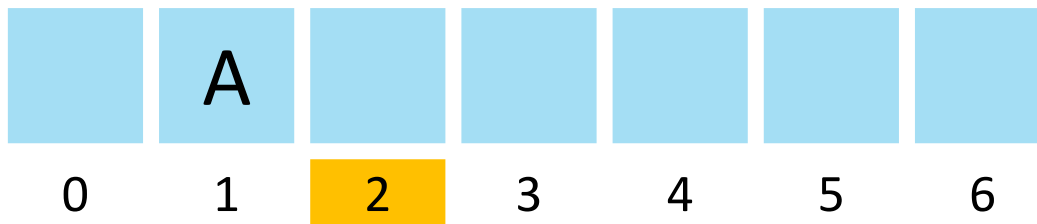
# Encadenamiento

$m = 7$ . Insertemos la A.  $h(A) = 15$ ;  $15 \bmod 7 = 1$



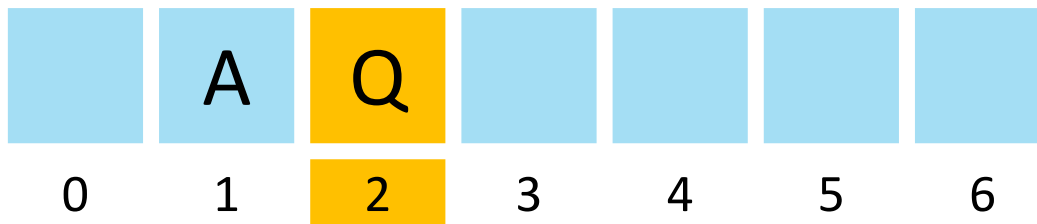
# Encadenamiento

$m = 7$ . Insertemos la Q.  $h(Q) = 37$ ;  $37 \bmod 7 = 2$



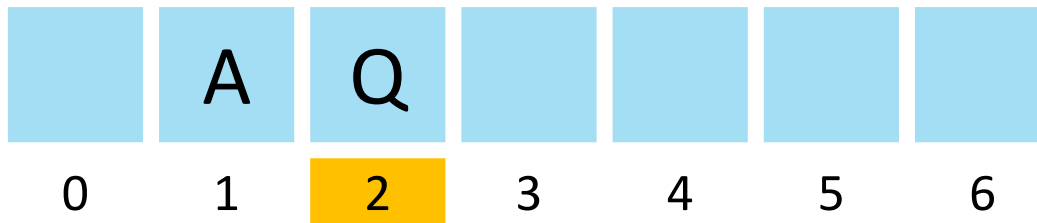
# Encadenamiento

$m = 7$ . Insertemos la Q.  $h(Q) = 37$ ;  $37 \bmod 7 = 2$



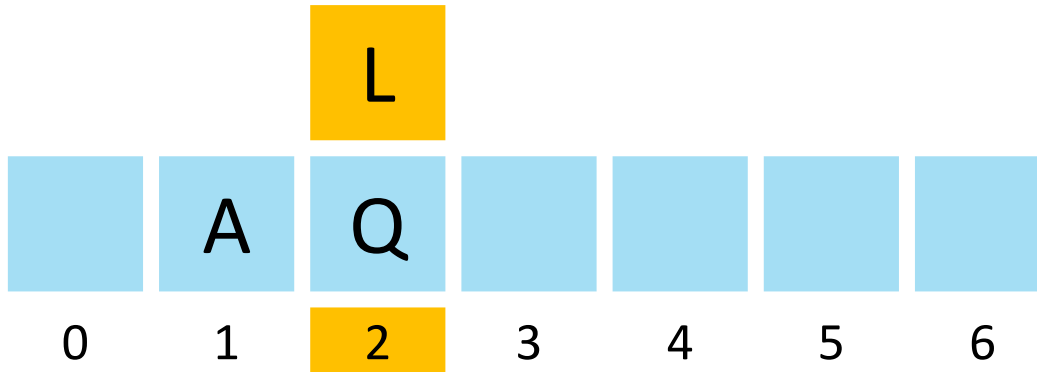
# Encadenamiento

$m = 7$ . Insertemos la L.  $h(L) = 51$ ;  $51 \bmod 7 = 2$



# Encadenamiento

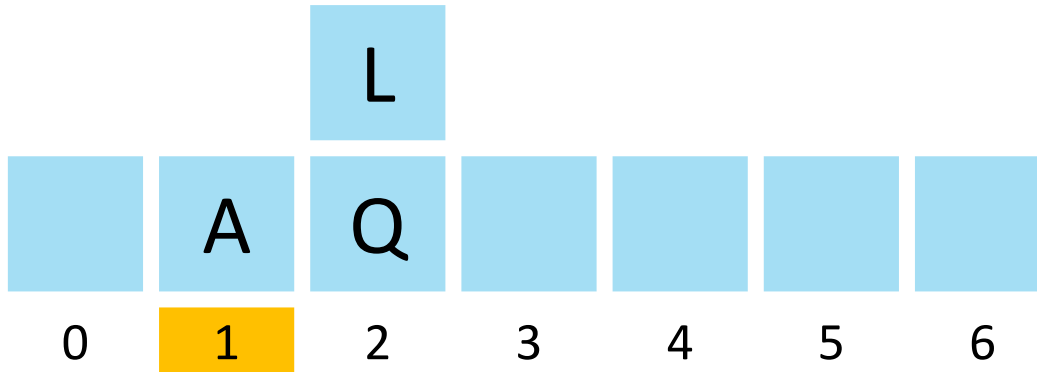
$m = 7$ . Insertemos la L.  $h(L) = 51$ ;  $51 \bmod 7 = 2$





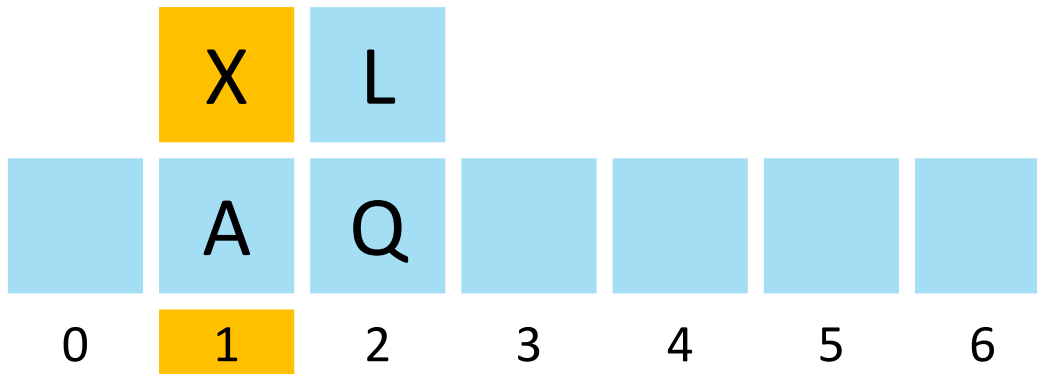
# Encadenamiento

$m = 7$ . Insertemos la X.  $h(X) = 29$ ;  $29 \bmod 7 = 1$



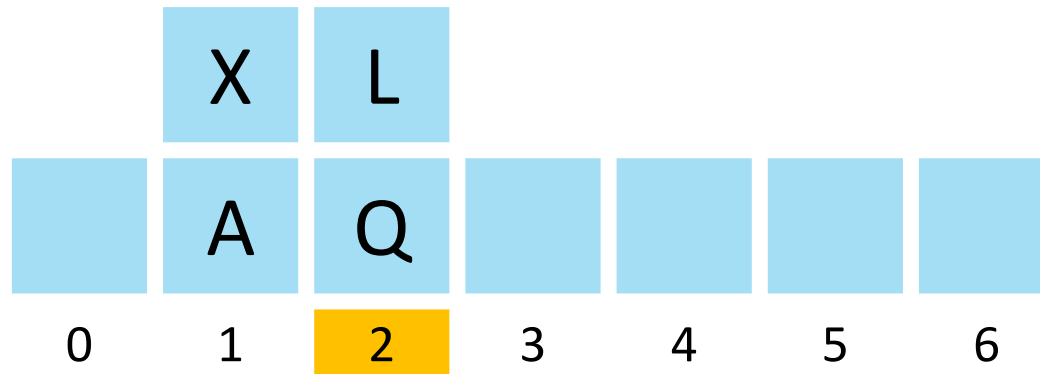
# Encadenamiento

$m = 7$ . Insertemos la X.  $h(X) = 29$ ;  $29 \bmod 7 = 1$



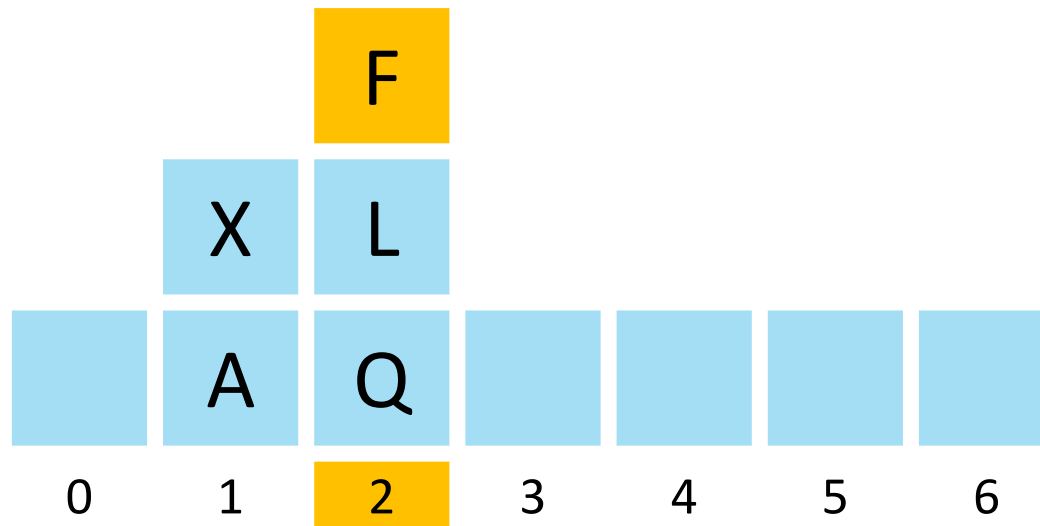
# Encadenamiento

$m = 7$ . Insertemos la F.  $h(F) = 58$ ;  $58 \bmod 7 = 2$



# Encadenamiento

$m = 7$ . Insertemos la F.  $h(F) = 58$ ;  $58 \bmod 7 = 2$



# Factor de Carga



Se define el factor de carga  $\lambda$  como:

$$\lambda = \frac{n}{m}$$

Podemos fijar un valor  $\lambda_{max}$ , y garantizar siempre que

$$\lambda < \lambda_{max}$$

# Rehashing



Si  $\lambda < \lambda_{max}$ , en algún momento hay que hacer crecer la tabla

A este proceso se le dice **rehashing** ¿Cuál es su complejidad?

¿Qué complejidad tendrían ahora las operaciones de la tabla?