Algoritmos de Remoción de Páginas

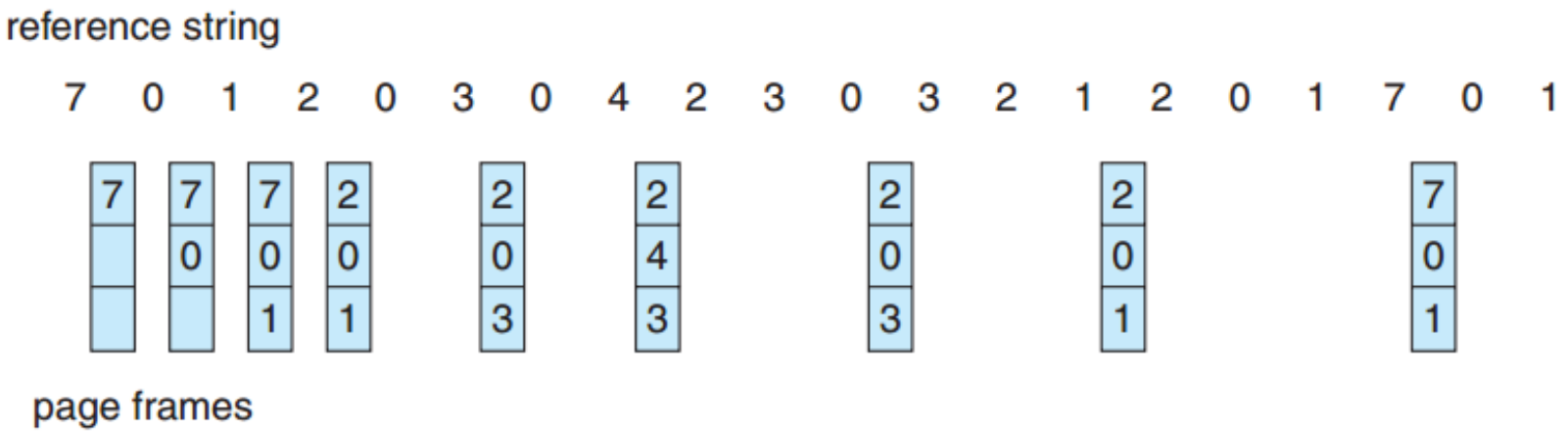
Victor Tortolero, 24.569.609

Sistemas Operativos, FACYT

29 de mayo de 2016

Algoritmo: Óptimo

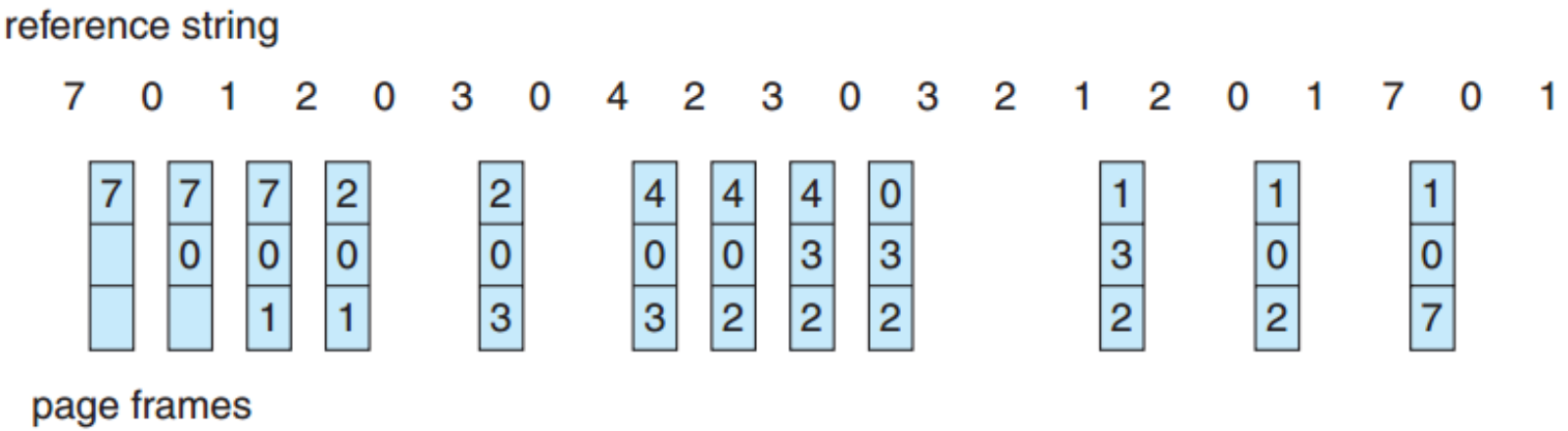
Cada pagina se etiqueta con el numero de instrucciones que se ejecutaran antes de que se referencie. La que tenga la etiqueta mas alta se eliminara al ocurrir un fallo de pagina.



Algoritmo: Least Recently Used (LRU)

Descarta la página que no se haya utilizada durante la mayor longitud de tiempo.

Implementación con Contador: Con un contador se asocia a cada pagina el tiempo de la ultima vez que se referencio.

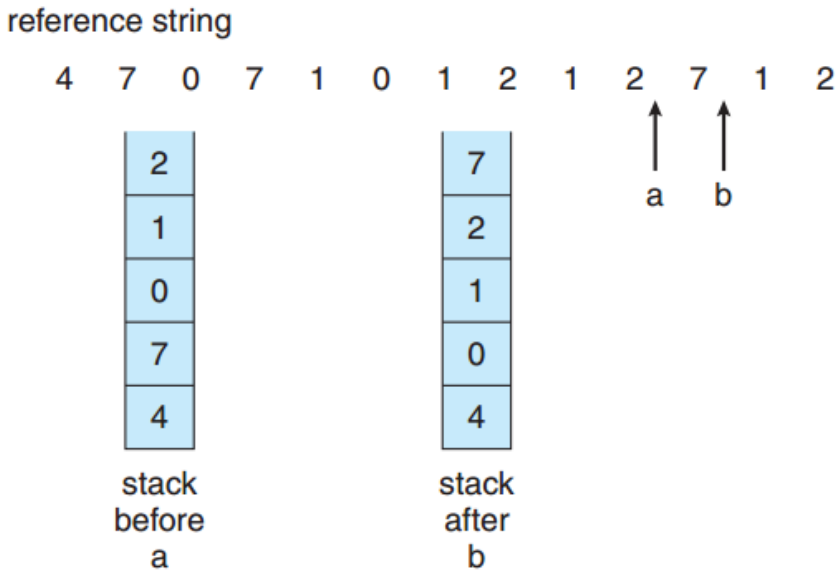


Algoritmo LRU con implementado con un contador

Algoritmo: Least Recently Used (LRU)

Descarta la página que no se haya utilizada durante la mayor longitud de tiempo.

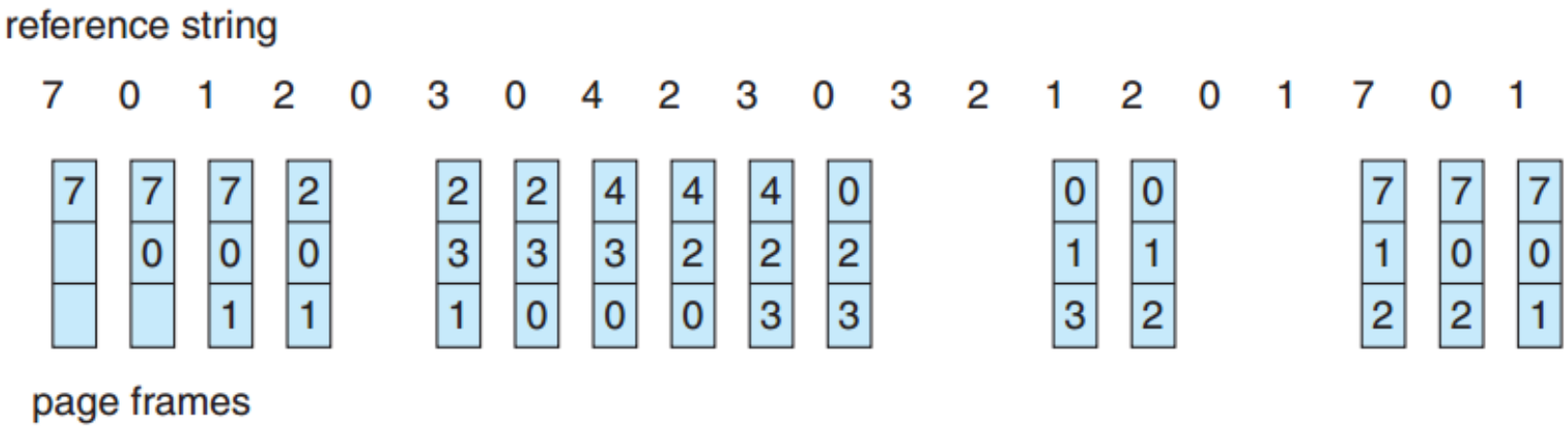
Implementación con Pila: Las paginas mas recientemente referenciadas se mantienen en el tope de la pila, y las menos en el fondo.



Algoritmo LRU implementado con una pila

Algoritmo: First in, First out (FIFO)

Se mantiene una lista de las paginas. Al ocurrir un fallo de pagina, se elimina la pagina que esta en la parte frontal y la nueva página se agrega a la parte final de la lista.

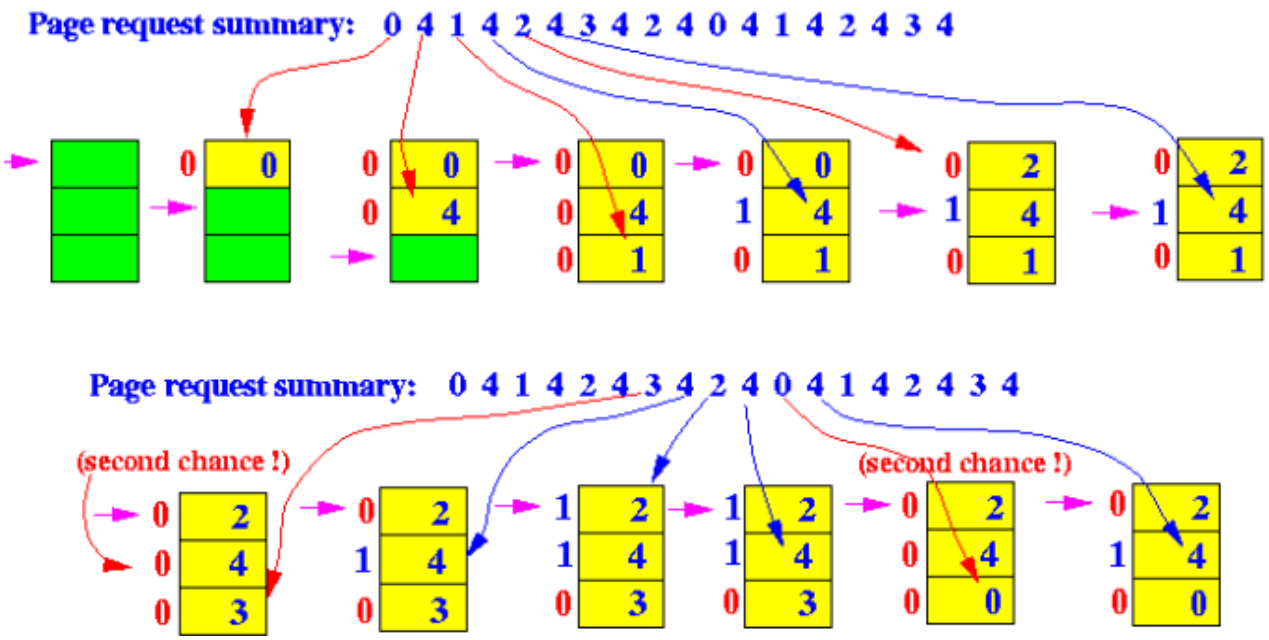


Algoritmo FIFO

Algoritmo: Segunda Oportunidad

Se revisa la pagina que esta al frente de una lista, y:

- Si $R = 0$, la pagina se substituye de inmediato.
- Si $R = 1$, se pone en 0, y la pagina se pasa al final de la lista como si acabara de llegar a memoria y sigue buscando.



Algoritmo Reloj

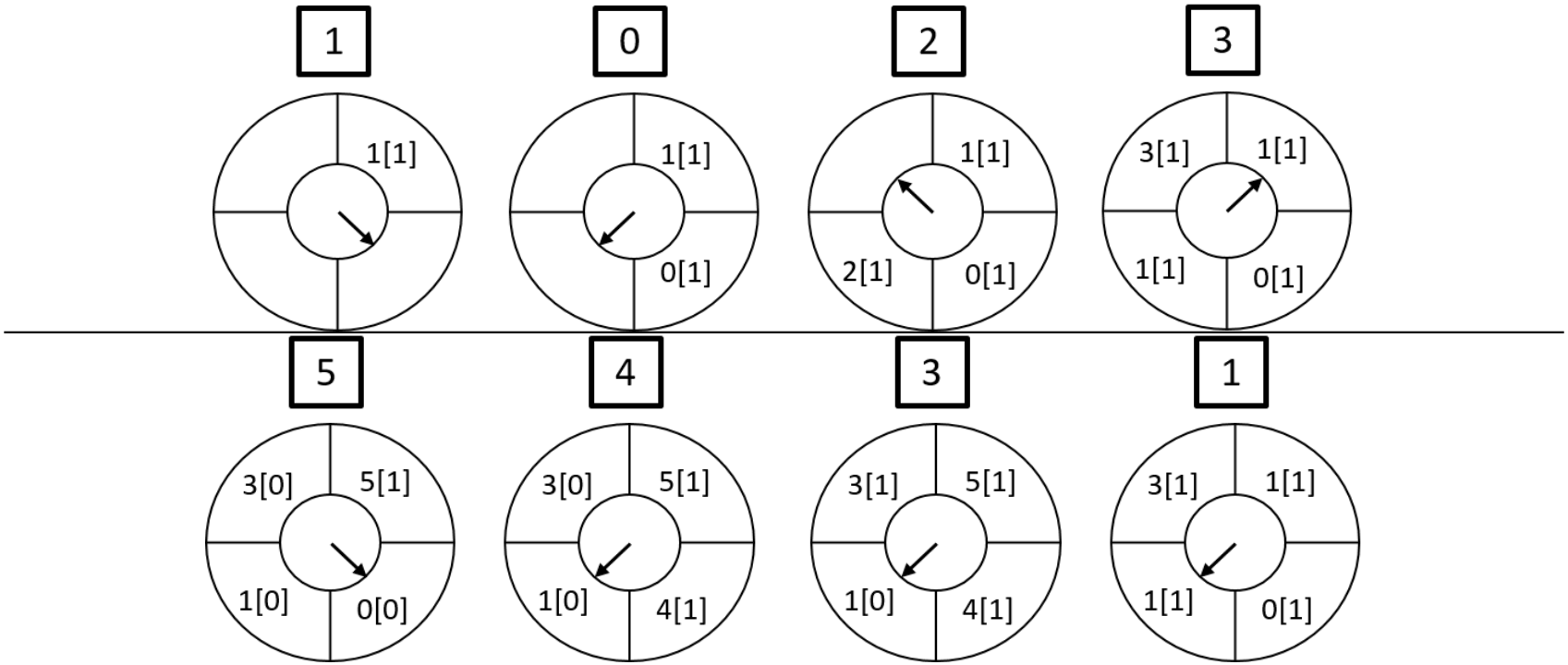
Algoritmo: Reloj

Las paginas se mantienen en una lista circular, y se apunta a una de ellas.

- Si $R = 0$, la pagina se substituye y avanza el apuntador.
- Si $R = 1$, se pone en 0, y se avanza el apuntador, y se repite hasta encontrar una pagina con $R = 0$.

Algoritmo: Reloj

Cadena de Referencia: 1 0 2 3 5 4 3 1 (#pagina[R])



Algoritmo: Not Recently Used (NRU)

A cada pagina se asocian 2 bits y se clasifican:

- **Clase 0**: no referenciada, no modificada ($R = 0$, $M = 0$).
- **Clase 1**: no referenciada, modificada ($R = 0$, $M = 1$).
- **Clase 2**: referenciada, no modificada ($R = 1$, $M = 0$).
- **Clase 3**: referenciada, modificada ($R = 1$, $M = 1$).

Al ocurrir fallo de pagina, se elimina una página al azar de la clase de menor numeración que no esté vacía.

Algoritmo: Working Set

Las paginas tienen una edad (tiempo virtual actual menos su tiempo de ultimo uso). Y se tiene un tamaño del conjunto de trabajo. Al ocurrir fallo de pagina, se recorre la tabla:

- Si $R = 1$ se actualiza el tiempo de ultimo uso con el tiempo virtual actual.
- Si $R = 0 \wedge edad > \tau$ se sustituye.
- Si $R = 0 \wedge edad \leq \tau$ se actualiza el tiempo de ultimo uso con el menor de los tiempos de ultimo uso.
- Si se recorre toda la tabla sin encontrar una victima:
  - Se sustituye la mas antigua con $R = 0$.
  - Si ninguna tiene $R = 0$, se sustituye una pagina al azar.

Algoritmo: Working Set

Ejemplo para la siguiente cadena de referencias con 3 marcos, y un reloj para borrar el bit R cada 4 pulsos, y con un $\tau = 2$:

Cadena de Referencia: 7 0 1 2 0 3 0

#pagina[R|tiempo de ultimo uso]

Tiempos	0	1	2	3*	4	5	6
Referencias	7	0	1	2	0	3	0
Marco 1	7[1|0]	7[1|0]	7[1|0]	2[1|3]	2[1|3]	2[1|5]	2[1|5]
Marco 2		0[1|1]	0[1|1]	0[0|2]	0[1|4]	0[1|5]	0[1|6]
Marco 3			1[1|2]	1[0|2]	1[0|2]	3[1|5]	3[1|5]
Fallos	X	X	X	X		X	

Algoritmo: WS Clock

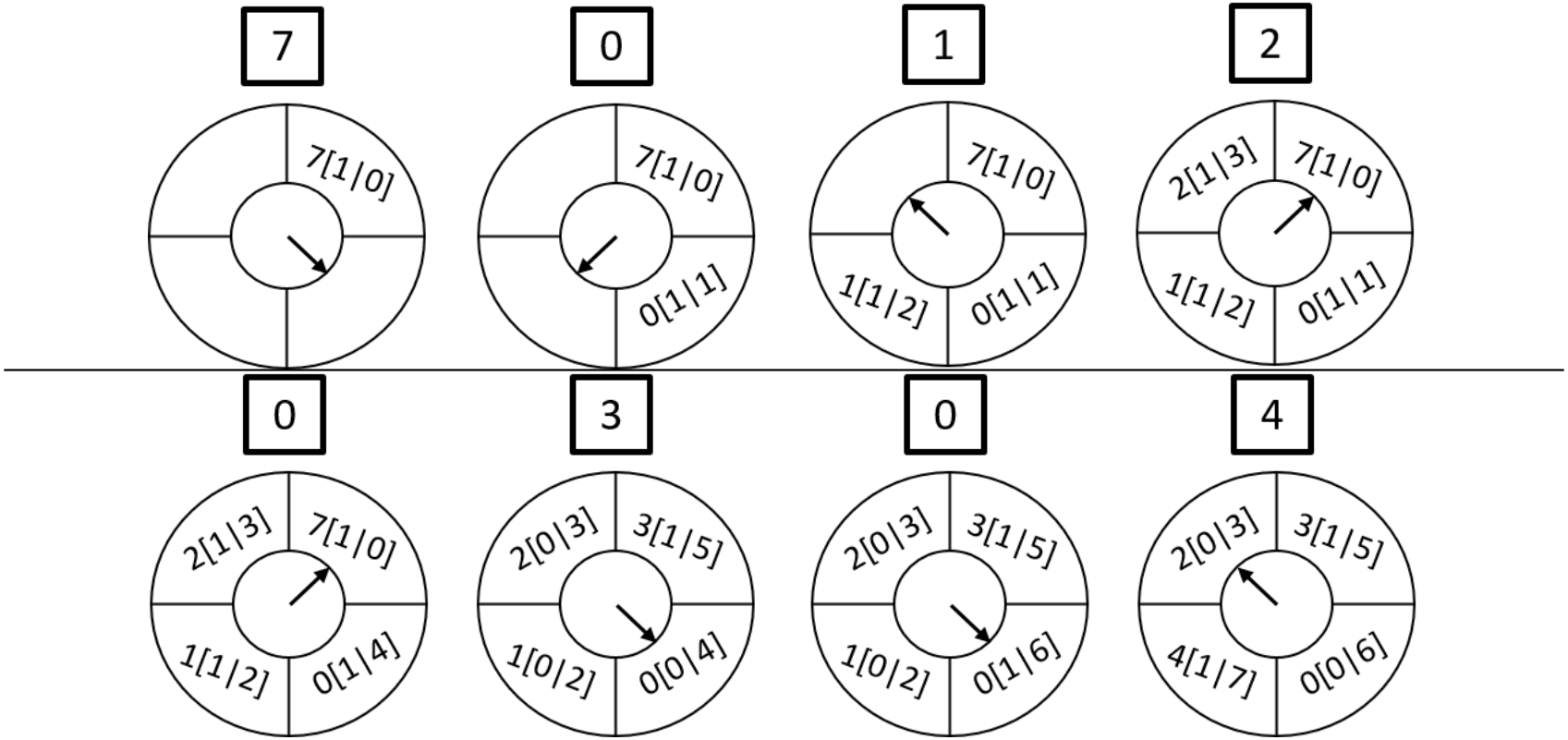
Mezcla entre el Working Set y el Clock.

- Si $R = 1$ se actualiza el tiempo de ultimo uso con el tiempo virtual actual, se pone en $R = 0$ y se avanza el apuntador.
- Si $R = 0$ y la edad $> \tau$ y la pagina esta limpia, se coloca en ese espacio.
- Si $R = 0$ y la edad $> \tau$ y la pagina esta sucia avanza el apuntador y sigue el algoritmo.

Algoritmo: WS Clock

Cadena de Referencia: 7 0 1 2 0 3 0 4

(#pagina[R|tiempo de ultimo uso])

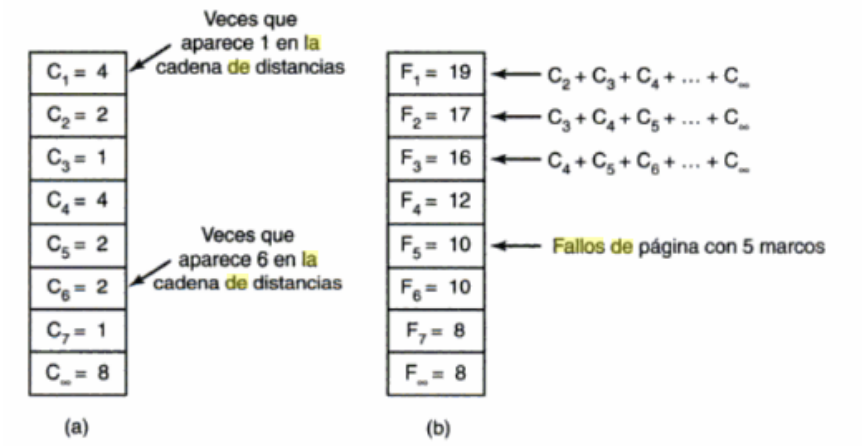


Prediccion de tasa de fallos

Se denota como C_i el numero de veces que aparece i. Luego calculamos con la formula

$$F_m = \sum_{k=m+1}^{n} C_k + C_\infty$$

El valor de F_m es el numero de fallos de pagina que se presentaran con la cadena de distancias dada y m marcos de pagina. C_∞ es el numero de veces que aparece ∞ en la cadena de distancias.



Anomalia de Belady

Es un efecto por el cual es posible tener mas fallos de pagina en el algoritmo FIFO al aumentar el numero de marcos en la memoria física. Antes de que se descubriera tal anomalía, se creía que al aumentar los marcos decrementaria los fallos de pagina.

Anomalia de Belady

Ejemplo con 3 marcos

Cadena de Referencia: 3 2 1 0 3 2 4 3 2 1 0 4

Referencias	3	2	1	0	3	2	4	3	2	1	0	4
Marco 1	3	3	3	0	0	0	4	4	4	4	4	4
Marco 2		2	2	2	3	3	3	3	3	1	1	1
Marco 3			1	1	1	2	2	2	2	2	0	0
Fallos	X	X	X	X	X	X	X			X	X	

9 Fallos

Anomalia de Belady

Ejemplo con 4 marcos

Cadena de Referencia: 3 2 1 0 3 2 4 3 2 1 0 4

Referencias	3	2	1	0	3	2	4	3	2	1	0	4
Marco 1	3	3	3	3	3	3	4	4	4	4	0	0
Marco 2		2	2	2	2	2	2	3	3	3	3	4
Marco 3			1	1	1	1	1	1	2	2	2	2
Marco 4				0	0	0	0	0	0	1	1	1
Fallos	X	X	X	X			X	X	X	X	X	X

10 Fallos