
THE UNIVERSITY *of* EDINBURGH
informatics

# Operating Systems (INFR10079)

2023/2024 Semester 2

# Virtual Memory

(Replacement Algorithms and More)

abarbala@inf.ed.ac.uk

Chapter 10

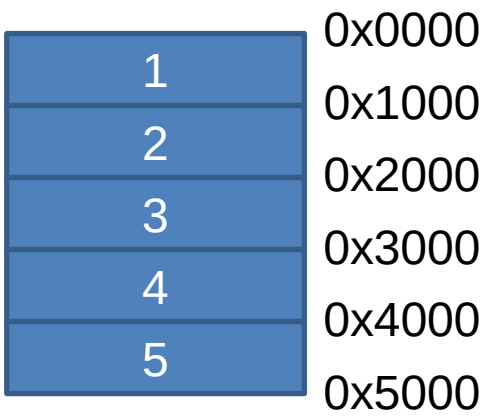
# Page Replacement Algorithm

- **What page to evict?**
  - Reduce page-fault rate by selecting **best victim page**
    - Reduce page-fault overhead
  - **Best victim page** is the one that will **never be touched again**
    - Not needed in the near future
  - **Belady's Theorem**
    - Evicting the page that won't be used for the longest period of time minimizes page fault rate
  - Evict **unmodified** pages first
    - **No need to write** them back to disk
- **Examine** page replacement algorithms
  - Assume that a process pages against itself
  - Using a fixed number of page frames

# String of Memory References

- Ordered list of pages the program will reference
  - Example 1, 2, 3, 4, 1, 2, 5, …

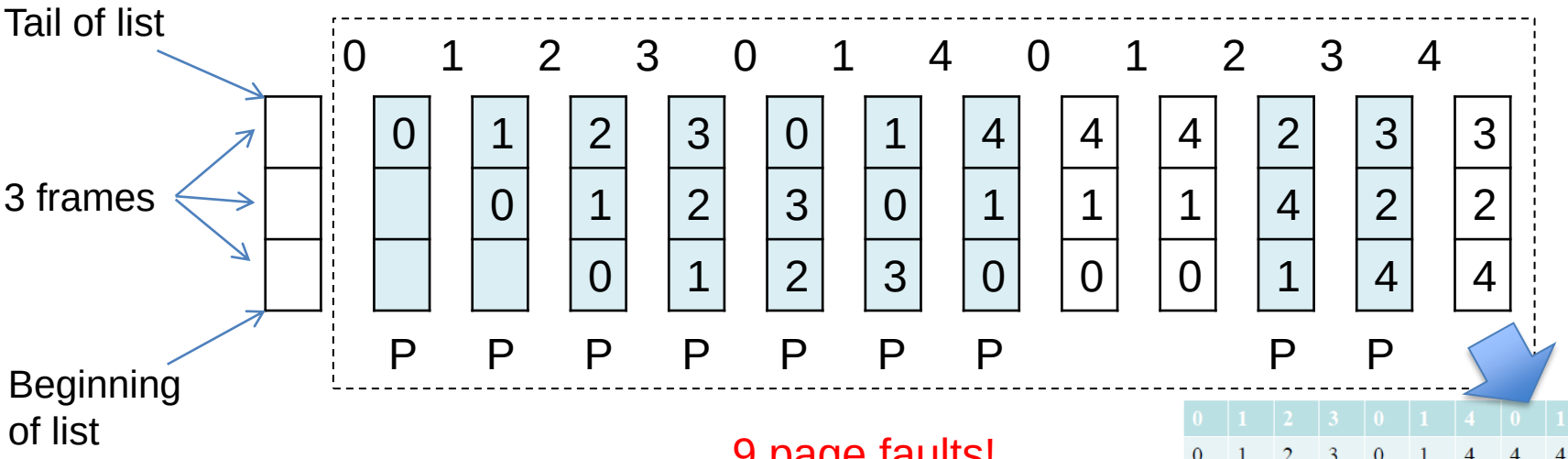
```
MOV R0, 0x0123
MOV R1, 0x1234
MOV R2, 0x2345
MOV R3, 0x3456
MOV 0x0100, R0
MOV 0x1200, R1
MOV R4, 0x4567
```



# First-In-First-Out (FIFO) Algorithm

- **Replace page that has been inserted first and is still in**

- 3 physical page frames, 5 virtual pages
- Reference string: **0, 1, 2, 3, 0, 1, 4, 0, 1, 2, 3, 4**

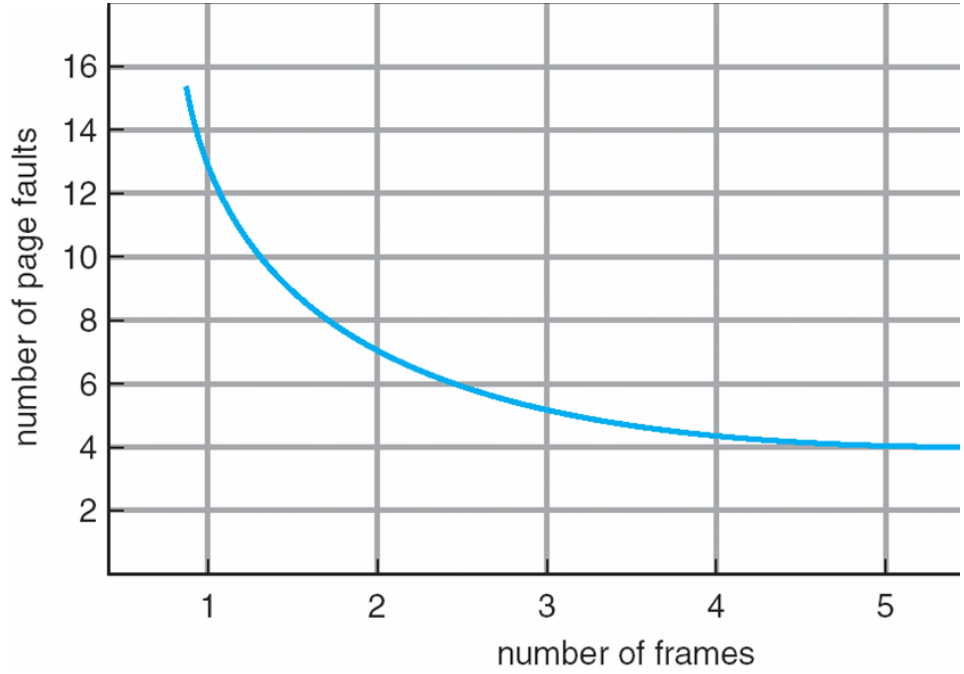



9 page faults!

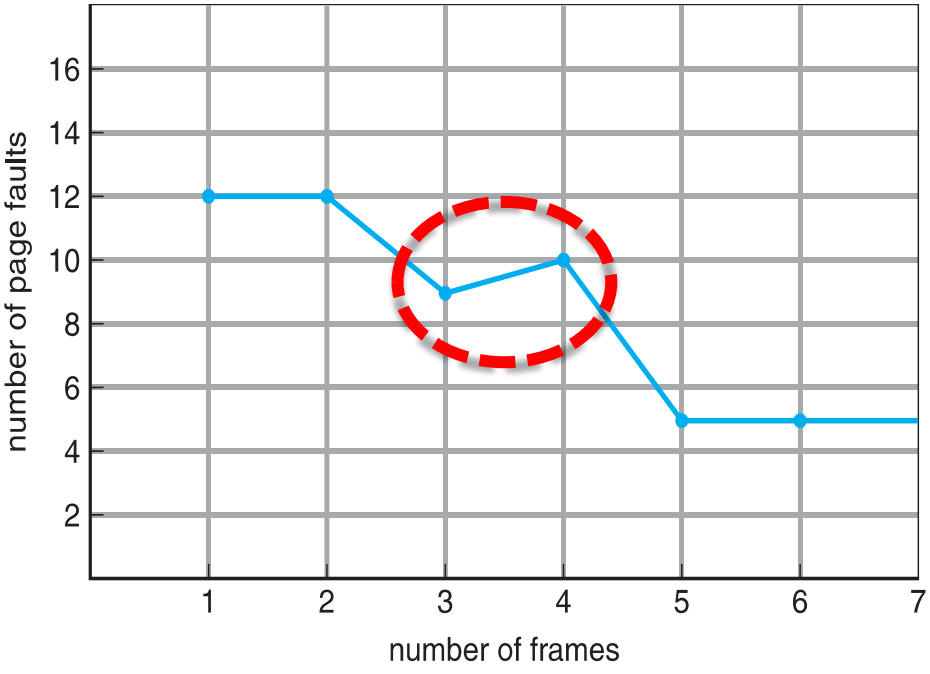
| 0 | 1 | 2 | 3 | 0 | 1 | 4 | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 0 | 1 | 4 | 4 | 4 | 2 | 3 | 3 |
| | 0 | 1 | 2 | 3 | 0 | 1 | 1 | 1 | 4 | 2 | 2 |
| | | 0 | 1 | 2 | 3 | 0 | 0 | 0 | 1 | 4 | 4 |
| P | P | P | P | P | P | P | | | P | P | |

- Easy to implement
  - Maintain a linked list of all pages in the order they come into memory

# Belady's Anomaly



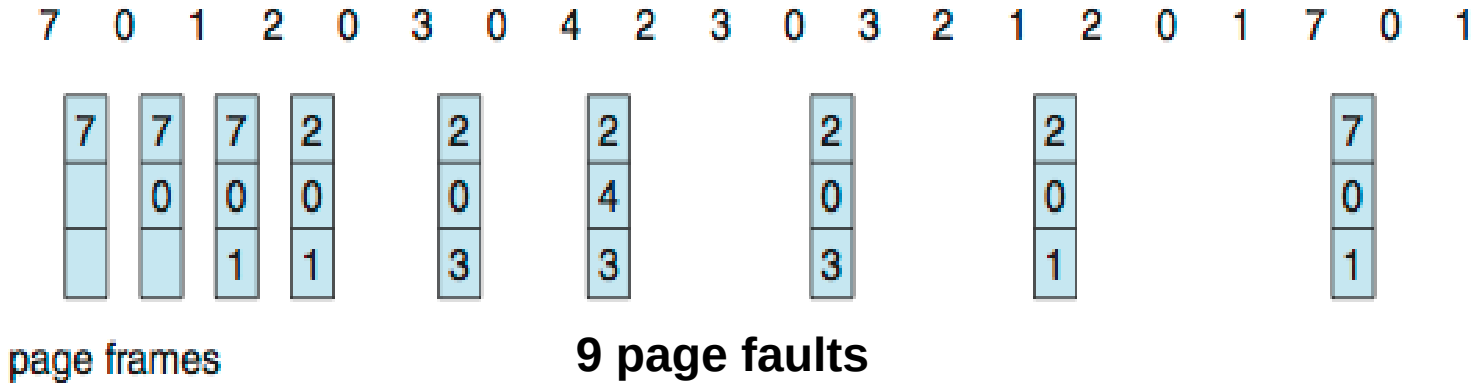

Expected Behavior
(more page frames less page faults)




FIFO Behavior
(more page frames do not guarantee less page faults)

# Optimal Algorithm

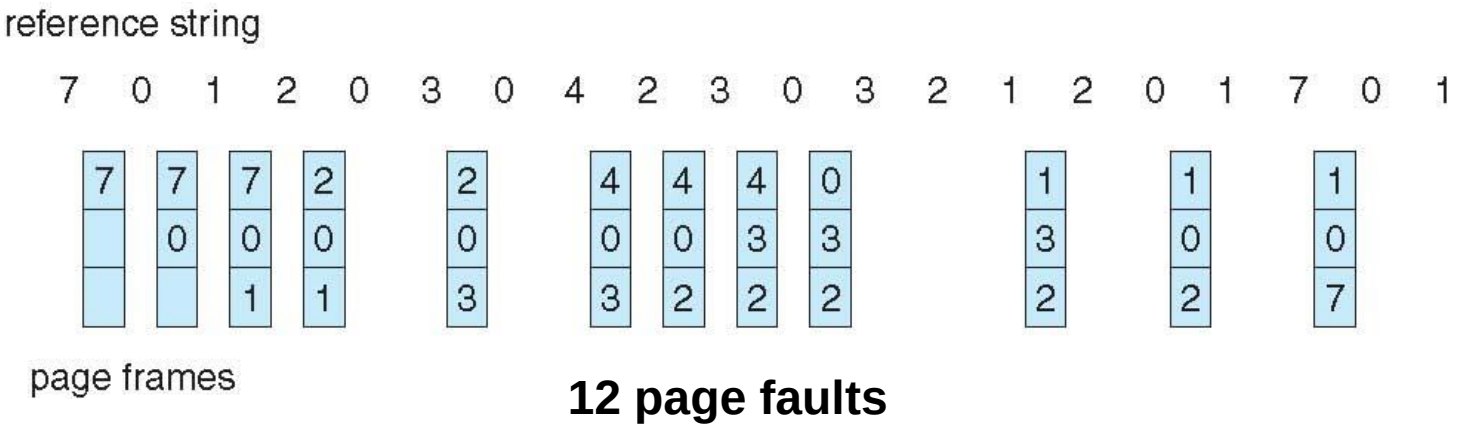
- **Replace page that will not be used for longest period**
  - Lowest page-fault rate
  - Never suffer from Belady's anomaly
- 3 physical page frames, 8 virtual pages
- Reference string: **7,0,1,2,0,3,0,4,2,3,0,3,0,3,2,1,2,0,1,7,0,1**




- How do you know what page will not be used?
  - **Can't read the future**
- Used for measuring how well your algorithm performs

# Least Recently Used (LRU) Algorithm

- **Replace page that has not been used in the most amount of time**
  - Use past knowledge rather than future
  - Never suffer from Belady's anomaly (stack algorithm)
- 3 physical page frames, 8 virtual pages
- Reference string: **7,0,1,2,0,3,0,4,2,3,0,3,0,3,2,1,2,0,1,7,0,1**




**12 page faults**

- Generally good algorithm and frequently used
- How to implement?
  - Associate time of last use with each page
  - Requires substantial **hardware assistance**

# Approximating LRU

- Use **page table entry bits**, maintained **by hardware**
  - Page referenced (if accessed or not)
  - Page modified (if access was in write)
- Keep a history/counter **for each page, in software**

- **History-based** page replacement algorithms
  - Recording the reference bits at regular intervals
    - keep history bits in a table in memory
  - *Aging*
  - *Second-chance (clock)*
  - *Enhanced second-chance*
- **Counting-based** page replacement algorithms
  - Keep a counter of the number of references that have been made
  - *Least frequently used (LFU)*
    - the page with the smallest count be replaced
  - *Most frequently used (MFU)*
    - the page with the highest count be replaced

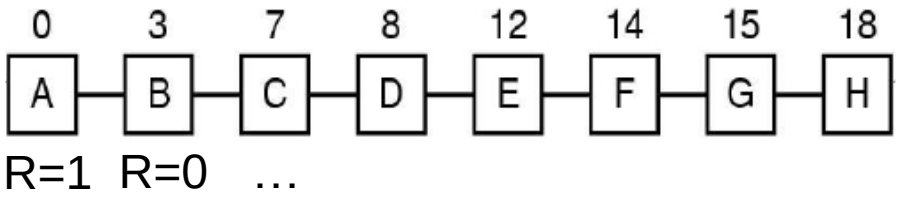
# Second Chance

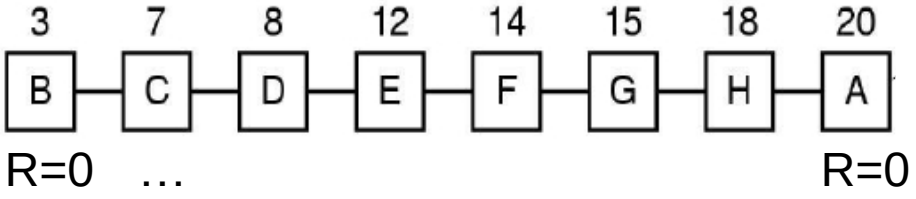
- FIFO variant
  - Adds the concept of usage (references)

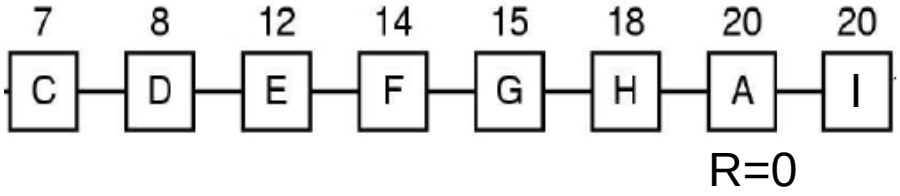
- Examine pages in FIFO order starting from beginning of list
  - Consider “reference bit”, R=0 **has not** been referenced
    a) IF R=0, remove page, go to **c)**
    b) IF R=1, set R=0 and place it at the end of FIFO list (hence, the second chance), go to **a)**
    c) Add new page at the end of FIFO (with R=1)
  - If not enough replaces, revert to pure FIFO on second pass

# Second Chance: Example

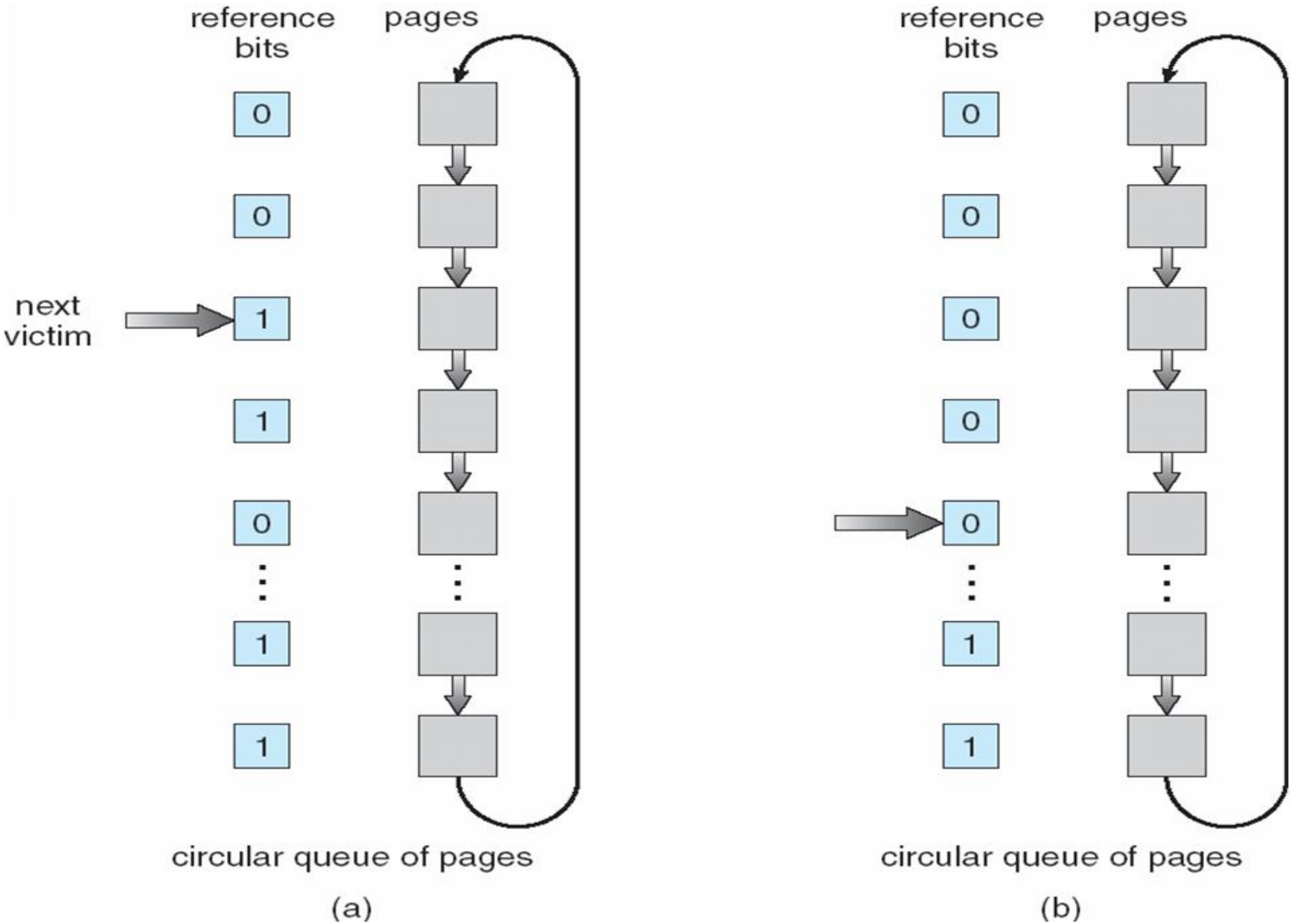
- We would like to insert [I]



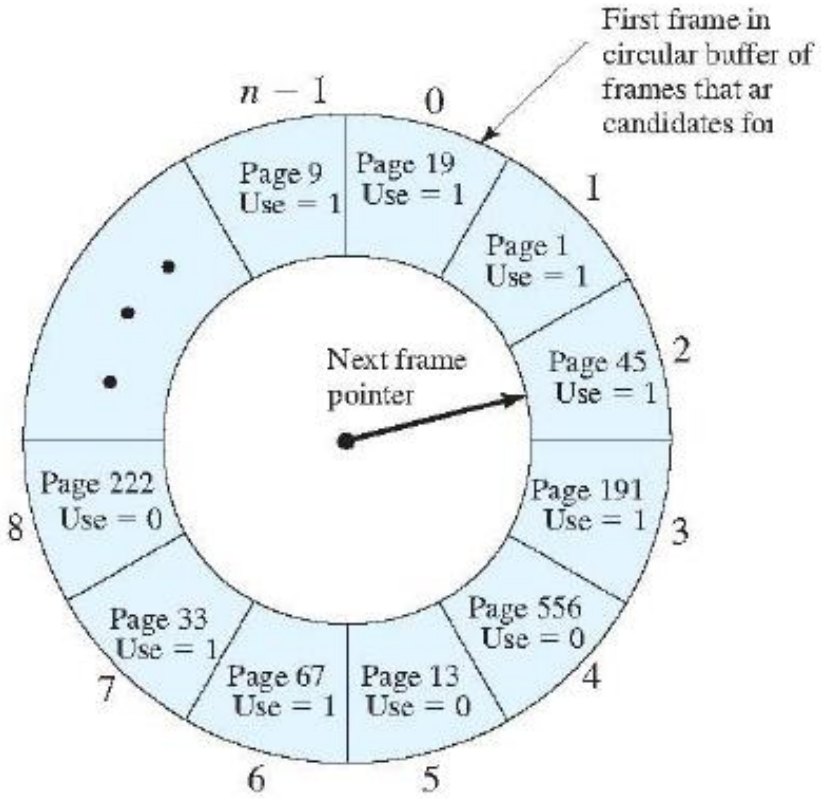



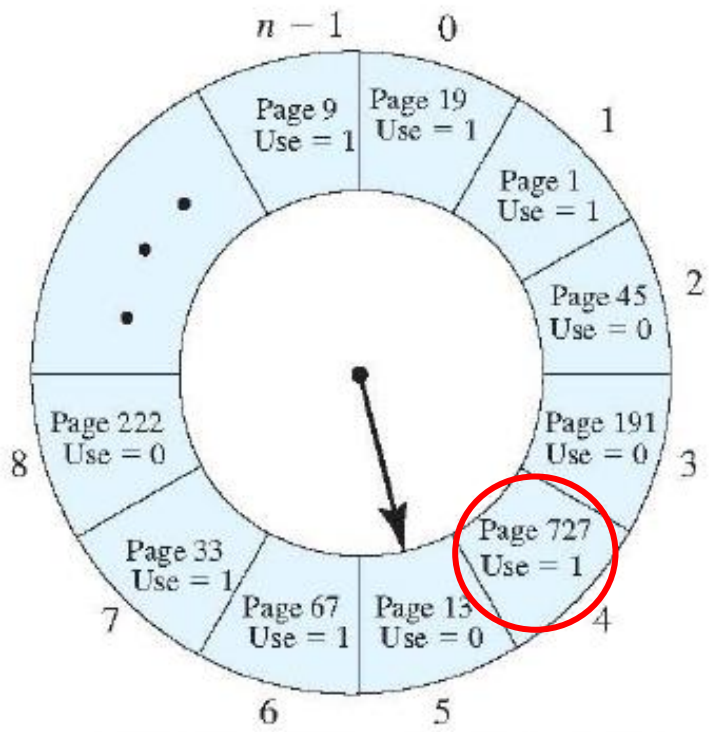
# Second Chance Clock



# Second Chance Clock: Example




(a) State of buffer just prior to a page replacement




(b) State of buffer just after the next page replacement

"Use" = "Reference"