Republic of the Philippines
Tarlac State University
COLLEGE OF COMPUTER STUDIES
Tarlac City, Tarlac
Tel. No. (045) 6068173

A Visualization Tool for First-In, First-Out (FIFO), Least Recently Used (LRU), and Optimal (OPT) Page Replacement Algorithms

Submitted by:

Cordero, Margaret Kate T.

BSCS 3B

Submitted to:

Jo Anne G. Cura

Instructor

Date of Submission: May 21, 2025

Republic of the Philippines
Tarlac State University
COLLEGE OF COMPUTER STUDIES
Tarlac City, Tarlac
Tel. No. (045) 6068173

Table of Contents

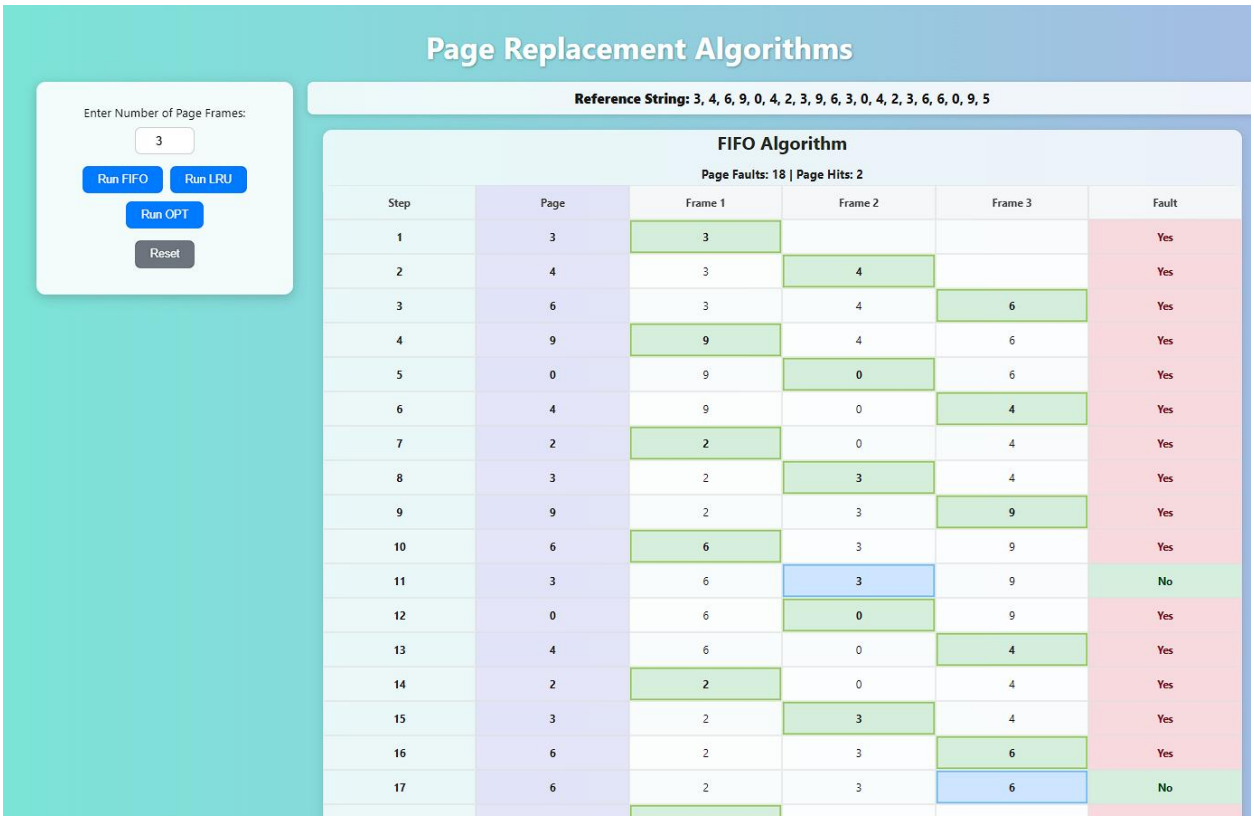
Introduction

Page replacement algorithms are a constantly difficult but essential subject in operating system development. Understanding the exact functions of algorithms like FIFO, LRU, and Optimal—that are, their mechanism to decide which page is removed to make room for a new one—can be challenging. It is crucial to how computers manage memory, especially when virtual memory is used to handle situations where a program requires more space than physically available RAM, frequently resulting in a "page fault." It is challenging to dynamically visualize the step-by-step procedure when relying just on static textbook diagrams or manual trace tables.

An interactive web-based page replacement algorithm simulator was created to fill this educational gap. This tool allows users to set the amount of memory frames and then watch as FIFO, LRU, or Optimal algorithms are executed using a produced page reference string. It performs this by using conventional web technologies, specifically HTML for structure, CSS for styling, and JavaScript for interactive logic. By allowing users to visually trace page faults and changes within memory frames in real-time, this development aimed to provide a more intuitive and engaging learning experience. It is believed that this simulator would help other students understand and grasp complicated operating systems principles.

Republic of the Philippines
Tarlac State University
COLLEGE OF COMPUTER STUDIES
Tarlac City, Tarlac
Tel. No. (045) 6068173

First In First Out (FIFO) Algorithm

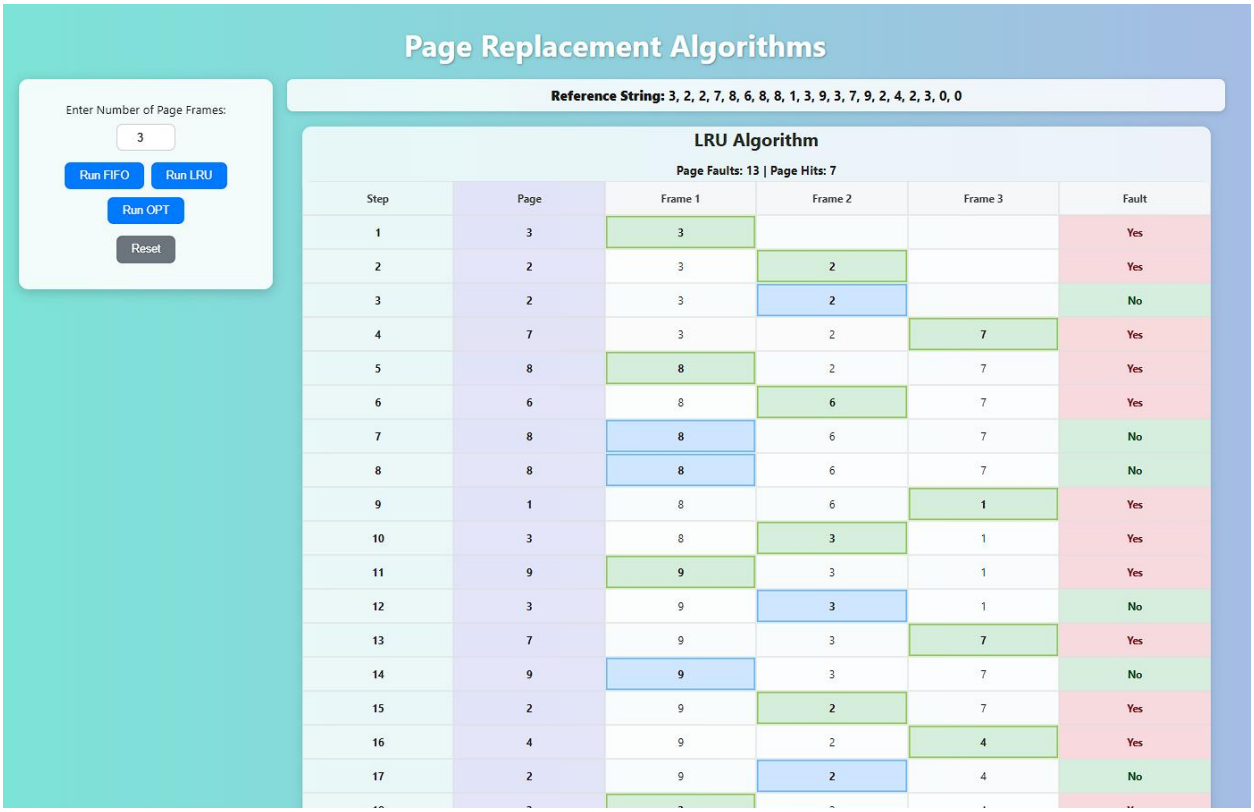


Step	Page	Frame 1	Frame 2	Frame 3	Fault
1	3	3			Yes
2	4	3	4		Yes
3	6	3	4	6	Yes
4	9	9	4	6	Yes
5	0	9	0	6	Yes
6	4	9	0	4	Yes
7	2	2	0	4	Yes
8	3	2	3	4	Yes
9	9	2	3	9	Yes
10	6	6	3	9	Yes
11	3	6	3	9	No
12	0	6	0	9	Yes
13	4	6	0	4	Yes
14	2	2	0	4	Yes
15	3	2	3	4	Yes
16	6	2	3	6	Yes
17	6	2	3	6	No

Figure 1: FIFO Algorithm Output

In this FIFO (First-In, First-Out) simulation, with a page reference string of 3, 4, 6, 9, 0, 4, 2, 3, 9, 6, 3, 0, 4, 2, 3, 6, 6, 0, 9, 5 the algorithm resulted in 18 page faults, 2 page hits and the number of page frames used is 3. One example of FIFO algorithm based on the image at step 5, Step 5, the computer needed page 2. It checked its memory frames (Frame 1: 9, Frame 2: 2, Frame 3: 5) and found that page 2 was already there in Frame 2. Because the page was already in memory, no page fault occurred, and the "Fault" column shows "No". The pages in memory were unaffected.

Least Recently Used (LRU) Algorithm

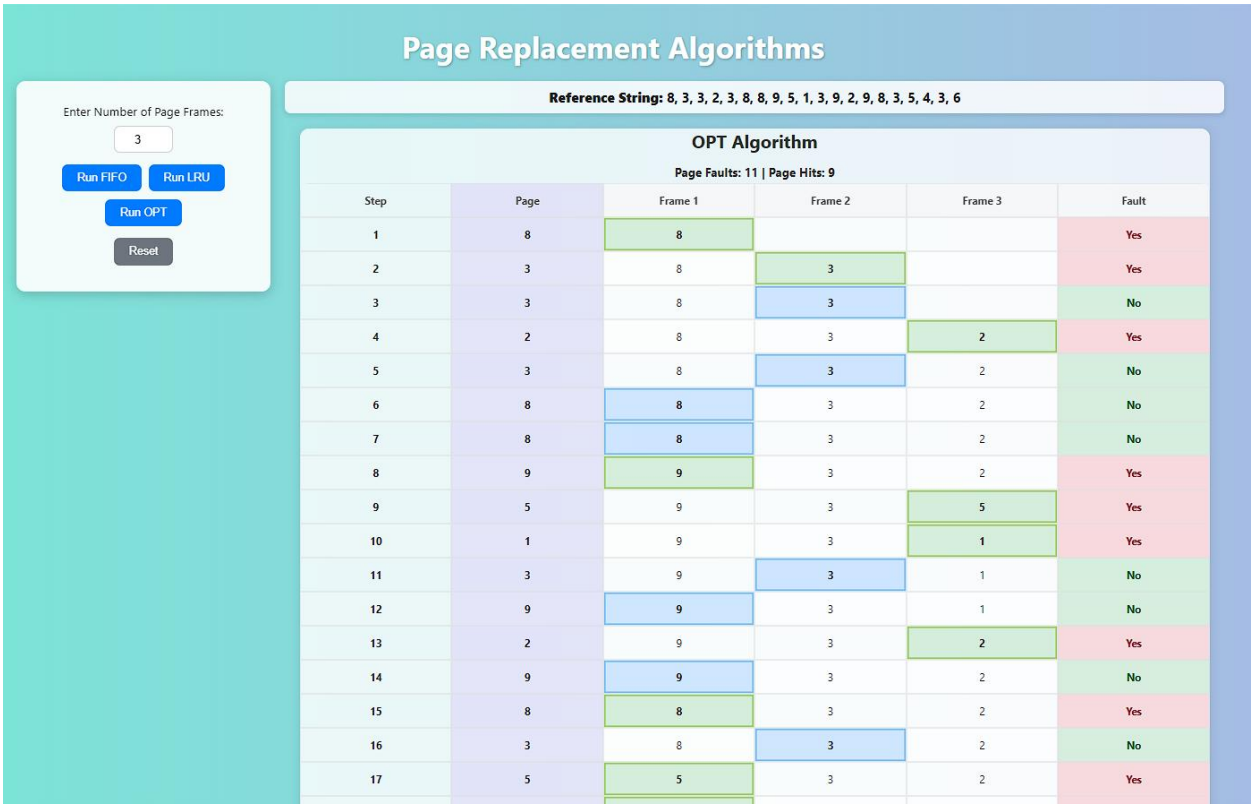


Step	Page	Frame 1	Frame 2	Frame 3	Fault
1	3	3			Yes
2	2	3	2		Yes
3	2	3	2		No
4	7	3	2	7	Yes
5	8	8	2	7	Yes
6	6	8	6	7	Yes
7	8	8	6	7	No
8	8	8	6	7	No
9	1	8	6	1	Yes
10	3	8	3	1	Yes
11	9	9	3	1	Yes
12	3	9	3	1	No
13	7	9	3	7	Yes
14	9	9	3	7	No
15	2	9	2	7	Yes
16	4	9	2	4	Yes
17	2	9	2	4	No

Figure 2: LRU Algorithm Output

In this LRU simulation, the page reference string is **3, 2, 2, 7, 8, 6, 8, 8, 1, 3, 9, 3, 7, 9, 2, 4, 2, 3, 0, 0** the page fault resulted in **13** faults, and page hits are **7**. From this LRU algorithm, at step 5 from the table the page requested '8'. At this point, the memory frames are full, containing pages '3', '2', and '7'. Since '8' is not in memory, a page fault occurs. In the LRU algorithm, we replace the page that has been least recently used. Looking back at the preceding steps, '7' was just used in Step 4, and '2' was used in Steps 2 and 3. However, '3' was used in Step 1 and then not touched again, making it the least recently used page. Therefore, '3' is removed from Frame 1 and is replaced by '8', resulting in the frames becoming '8', '2', '7'.

Optimal (OPT) Algorithm



Step	Page	Frame 1	Frame 2	Frame 3	Fault
1	8	8			Yes
2	3	8	3		Yes
3	3	8	3		No
4	2	8	3	2	Yes
5	3	8	3	2	No
6	8	8	3	2	No
7	8	8	3	2	No
8	9	9	3	2	Yes
9	5	9	3	5	Yes
10	1	9	3	1	Yes
11	3	9	3	1	No
12	9	9	3	1	No
13	2	9	3	2	Yes
14	9	9	3	2	No
15	8	8	3	2	Yes
16	3	8	3	2	No
17	5	5	3	2	Yes

Figure 3: OPT Algorithm Output

In this OPT simulation, the page reference string is **8, 3, 3, 2, 3, 8, 8, 9, 5, 1, 3, 9, 2, 9, 8, 3, 5, 4, 3, 6,** in this algorithm page faults resulted in **11** faults and the page hits is **9**. In this OPT algorithm, at step 8 of the table the page requested '9'. The memory frames currently contain pages '8', '3' and '2'. Since '9' is not in memory, a page fault occurs. For the Optimal (OPT) algorithm, we look into the future of the reference string to see which page currently in memory will be used furthest in the future, or not at all. In this specific step, '8' is not seen again in the immediate future of the reference string shown in the table. Therefore, '8' is chosen for replacement, and '9' is placed into Frame 1, resulting in the frames becoming '9', '3', '2'.

Conclusion

This project is a helpful tool for understanding how computers manage their fast, temporary memory, called "frames." When a computer needs data, it tries to find it in these frames. If it's not there, a "page fault" occurs, and the computer has to decide which old data to remove to make space for the new data.

The project lets you see three ways computers make these decisions. FIFO (First-In, First-Out) replaces the oldest data, just like a line. LRU (Least Recently Used) is smarter; it removes the data that hasn't been used for the longest time, guessing it's less likely to be needed soon.

Lastly, the Optimal algorithm shows the best possible scenario. It "knows" which data won't be needed for the longest time and removes that. While real computers can't do this, it helps us see how good the other methods are in comparison.