



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



A case study  
In partial fulfillment of the requirements  
for the course Operating System

Page Replacement Algorithms Simulator (FIFO,  
LRU and Optimal Algorithm)

---

Operating Systems Case Study

Submitted by:

**Dela Cruz, Rhed T.**

**BSCS-3B**

Submitted To:

**Ma'am Jo Anne G. Cura**

**May 21, 2025**



## Table of Contents

|                              |          |
|------------------------------|----------|
| Table of Contents.....       | i        |
| <b>I. DOCUMENTATION.....</b> | <b>1</b> |
| 1. User Interface.....       | 1        |
| 2. Computation Result.....   | 3        |
| 2.1 First Sample Input.....  | 3        |
| 2.2 Second Sample Input..... | 4        |
| 2.3 Third Sample Input.....  | 5        |



## I. DOCUMENTATION

### 1. User Interface

### Page Replacement Algorithm Simulator

Reference String:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

Number of Frames:

3

Calculate

#### Page Replacement Algorithm

A Page Replacement Algorithm is a method used in operating systems to manage memory when there is limited physical RAM available. It decides which memory pages to remove when a new page needs to be loaded, helping to minimize page faults and maintain efficient system performance. Common algorithms include FIFO (First-In, First-Out), LRU (Least Recently Used), and Optimal, each with its own strategy for selecting which page to replace. These algorithms are essential in virtual memory systems to ensure smooth execution of programs by optimizing memory usage.

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

The FIFO (First-In, First-Out) page replacement algorithm is a method used in computer systems to manage memory. It works by replacing the oldest page in memory with a new one when space is needed. The pages are arranged in a queue, and when a new page is requested, the page that has been in memory the longest is removed. While this algorithm is simple and easy to implement, it may not always be the most efficient, as it does not take into account how often or recently a page is used.

#### Least Recently Used (LRU) Algorithm

The LRU (Least Recently Used) page replacement algorithm is used in computer systems to manage memory by replacing the page that has not been used for the longest period of time. It keeps track of the order in which pages are accessed, and when a new page needs to be loaded into memory, it identifies and removes the page that was used least recently. This approach is more efficient than FIFO in many cases because it tends to keep the more frequently accessed pages in memory, reducing the number of page faults. However, it requires extra overhead to keep track of the access history of the pages.

#### Optimal Algorithm

The Optimal page replacement algorithm is considered the most efficient page replacement strategy, as it aims to minimize the number of page faults. It works by replacing the page that will not be used for the longest period of time in the future. The algorithm looks ahead at the future page references and chooses the page that will not be needed for the longest duration. While it guarantees the lowest page fault rate, it is not practical for real systems because it requires knowledge of future memory accesses, which is generally not available. However, it serves as a benchmark for evaluating other algorithms.

Created by Rhed Dela Cruz, Student at Tarlac State University. Developed as part of academic work in Operating System.

Figure 1: Popup Interface

Figure 1 shows the popup interface of the application, which includes textboxes for the reference string and number of frames, both pre-filled with default input values to guide users. It also displays information about the selected page replacement algorithm. This interface helps users better understand how the algorithm processes input and calculates page faults.

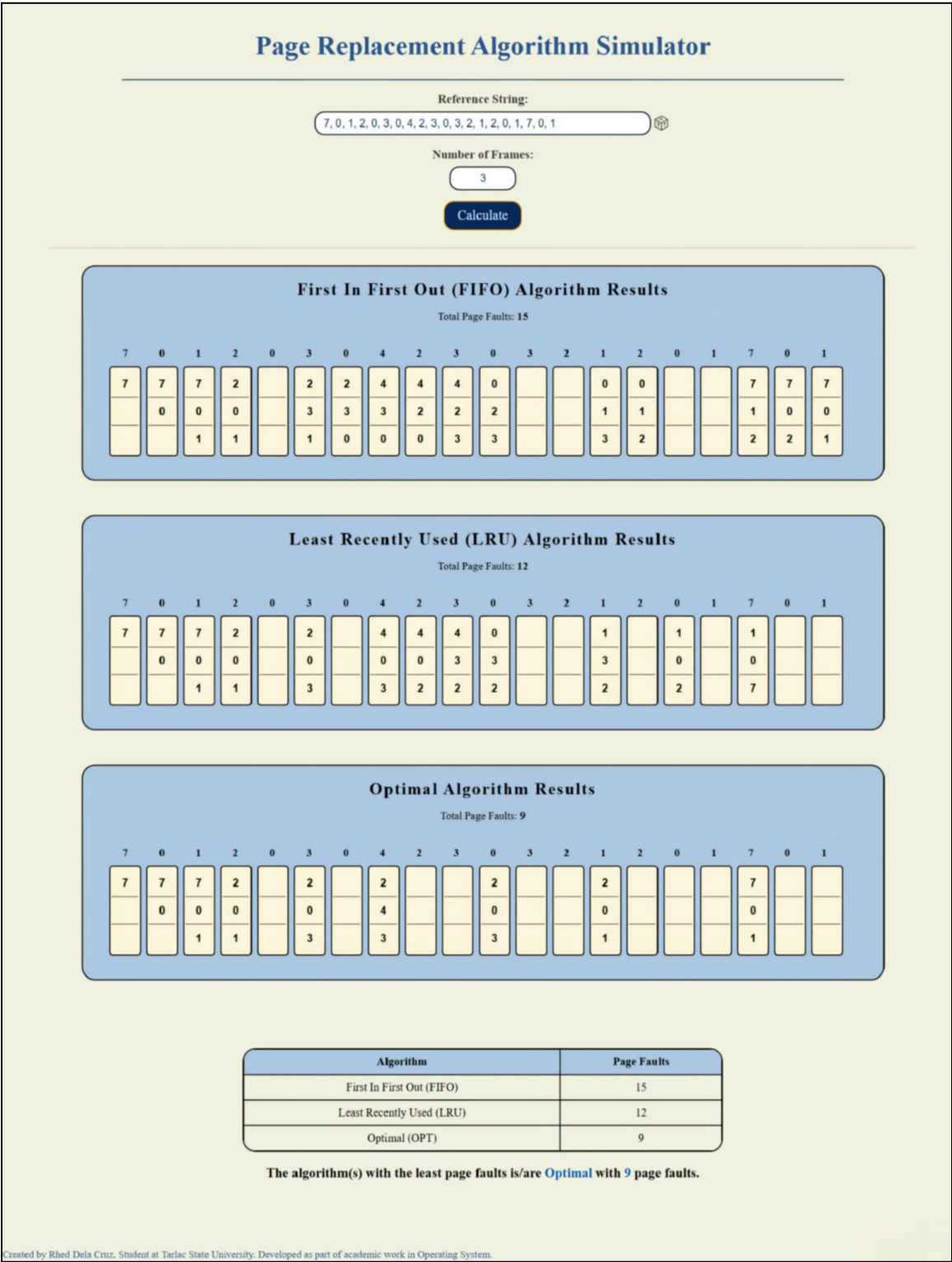


Figure 2: Computation Interface

Figure 2 shows the computation interface, where the results of the page replacement algorithm are displayed, including the number of page faults and the steps of the process based on the user's input.



## 2. Computation Result

### 2.1 First Sample Input

Reference String: 2, 0, 3, 4, 6, 7, 0, 0, 2, 5, 1, 2, 9, 5, 2, 5, 8, 1, 6, 4

Number of Frames: 3

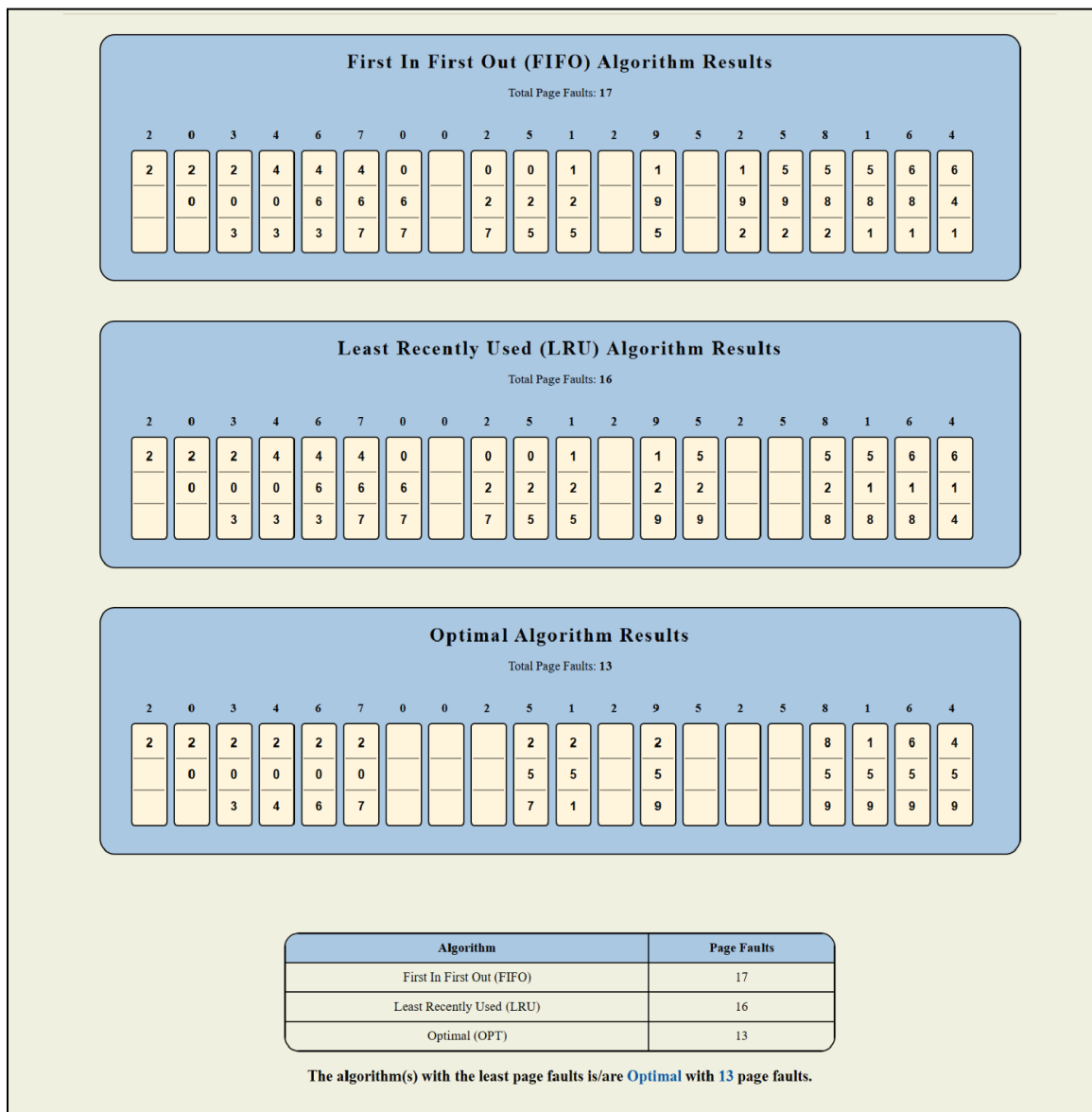


Figure 3: First Input Results

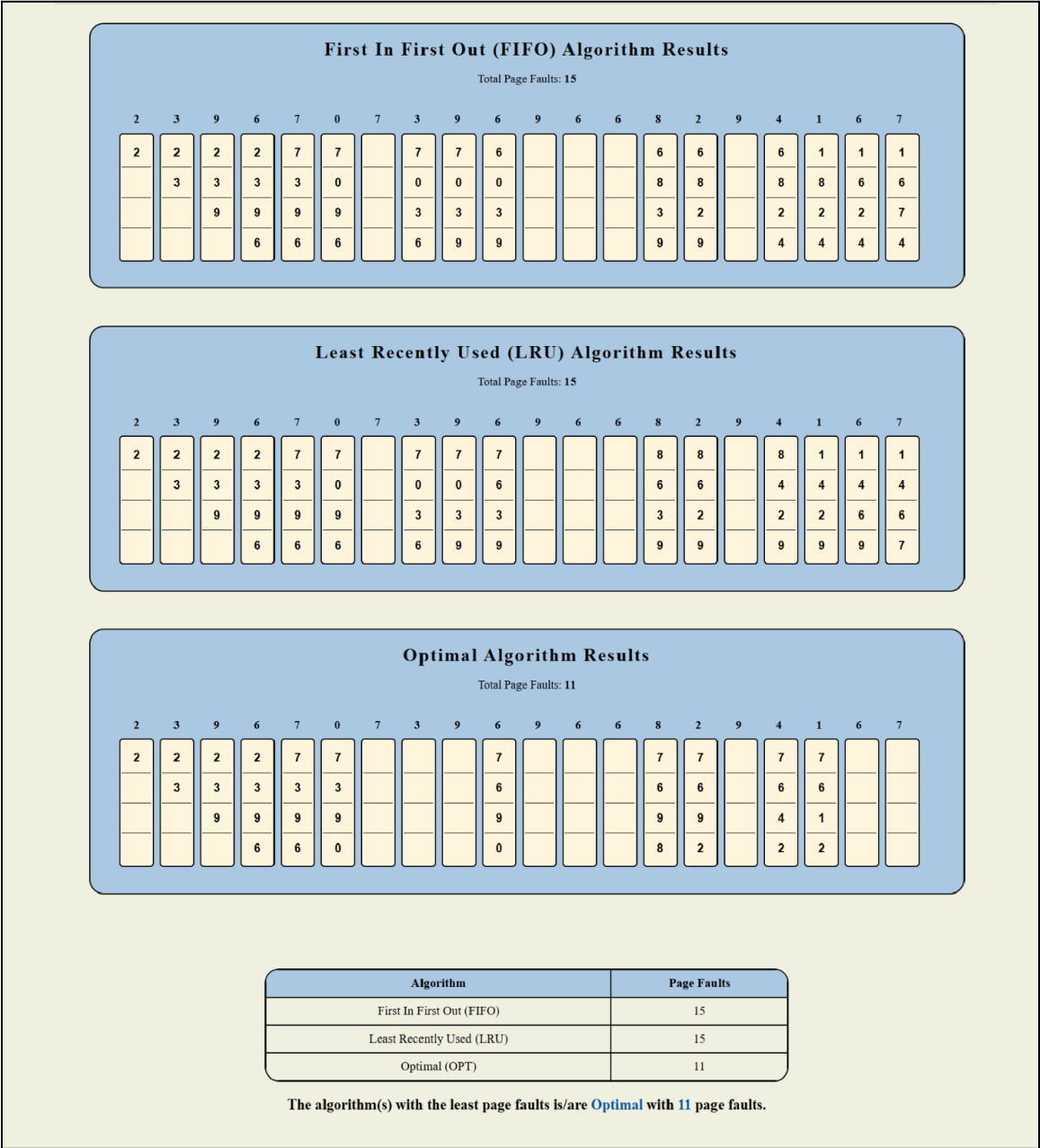
In this case, the Optimal Algorithm resulted in the least page faults (13). The Least Recently Used (LRU) Algorithm had a few more page faults (16), while the First-In-First-Out (FIFO) Algorithm had the highest number of page faults (17).



## 2.2 Second Sample Input

Reference String: 2, 3, 9, 6, 7, 0, 7, 3, 9, 6, 9, 6, 6, 8, 2, 9, 4, 1, 6, 7

Number of Frames: 4





### 2.3 Third Sample Input

Reference String: 7, 0, 2, 4, 3, 1, 8, 5, 3, 9, 8, 8, 3, 2, 1, 6, 4, 1, 2, 9

Number of Frames: 5

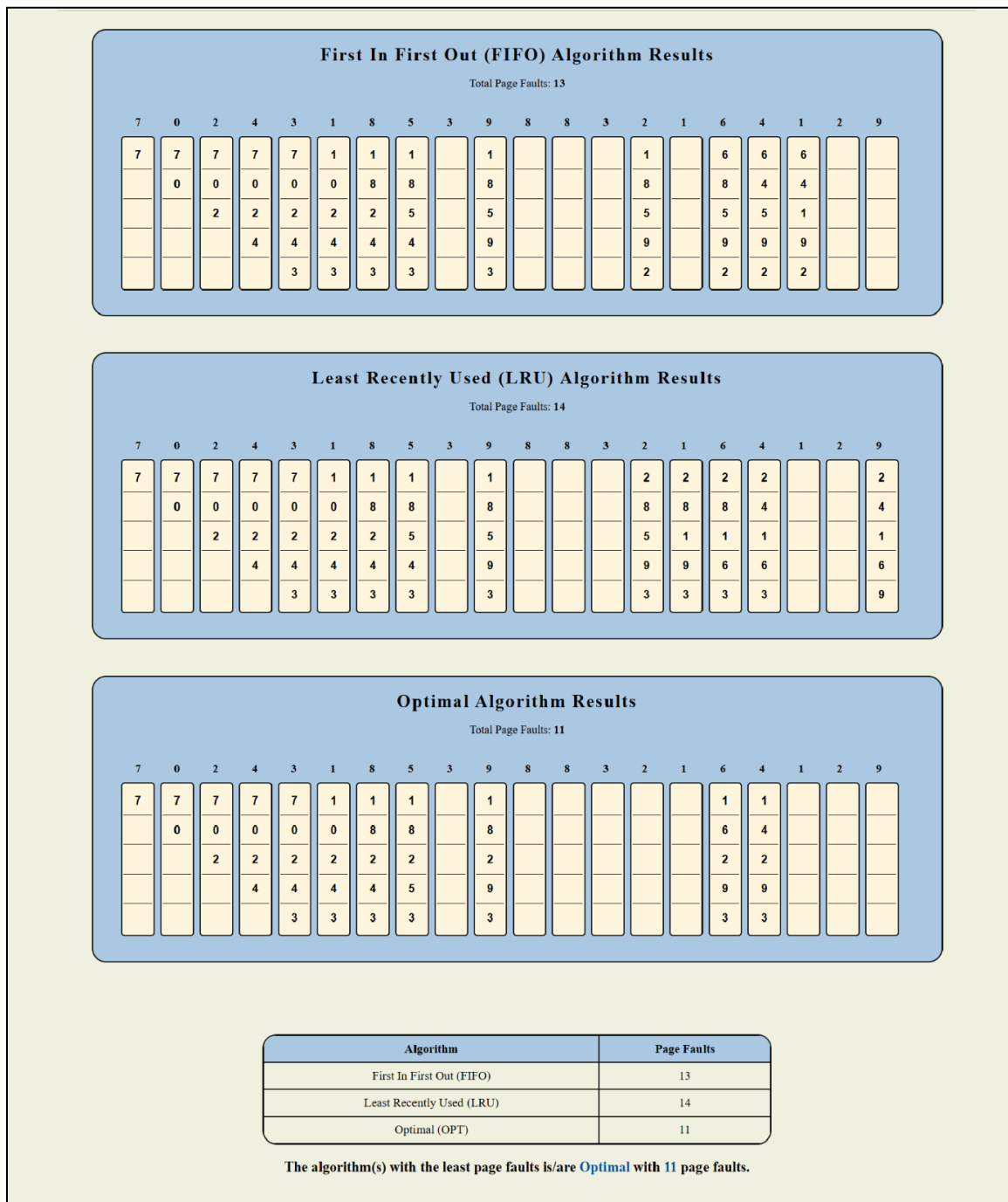


Figure 5: Third Input Results

In this case, the Optimal Algorithm resulted in the least page faults (11). The First-In-First-Out (FIFO) Algorithm had slightly more page faults (13), while the Least Recently Used (LRU) Algorithm had the highest number of page faults (14).