Name : Subham
Class : CSE(2)
Roll no : 2023UCS1603
Subject : OS-Lab

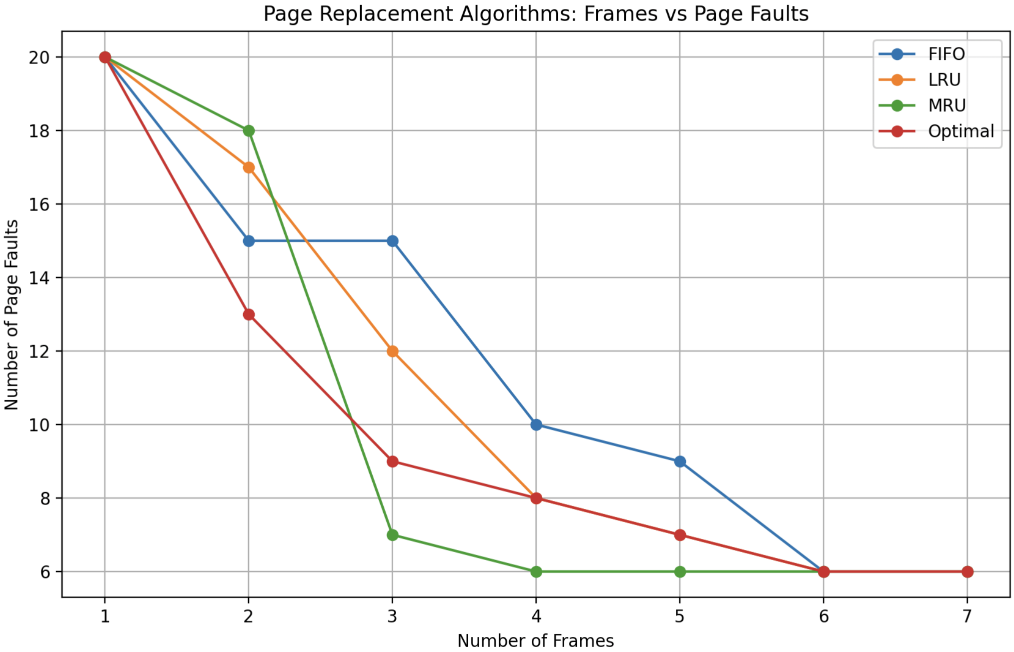
Question) Page replacement algorithms: LRU, MRU, FIFO, Optimal. in this i want a graph between no of frames vs no of page faults for the reference string and take a reference string..

Code :

```
import matplotlib.pyplot as plt
from collections import deque

# Page replacement algorithms
def fifo(ref_string, frames):
    memory = []
    page_faults = 0
    for page in ref_string:
        if page not in memory:
            page_faults += 1
            if len(memory) >= frames:
                memory.pop(0)
            memory.append(page)
    return page_faults

def lru(ref_string, frames):
    memory = []
    page_faults = 0
    for page in ref_string:
        if page not in memory:
            page_faults += 1
            if len(memory) >= frames:
                # Remove least recently used page
                memory.pop(0)
        else:
            # Move the page to the end of memory to show
it was recently used
            memory.remove(page)
        memory.append(page)
    return page_faults

def mru(ref_string, frames):
    memory = []
    page_faults = 0
    for page in ref_string:
        if page not in memory:
            page_faults += 1
            if len(memory) >= frames:
                # Remove the most recently used page
                memory.pop(-1)
        memory.append(page)
    return page_faults

def optimal(ref_string, frames):
    memory = []
    page_faults = 0
    for i in range(len(ref_string)):
        page = ref_string[i]
        if page not in memory:
            page_faults += 1
            if len(memory) >= frames:
                # Find the page with the farthest next use
                farthest, index = -1, -1
                for m in memory:
                    if m not in ref_string[i + 1:]:
                        index = memory.index(m)
                        break
                    else:
                        next_use = ref_string[i +
1:].index(m)
                        if next_use > farthest:
                            farthest = next_use
                            index = memory.index(m)
                memory.pop(index)
            memory.append(page)
    return page_faults

# Reference string and frame range
ref_string
```

```
= [7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7,
0, 1]
frame_range = range(1, 8)  # Testing for frame sizes from
1 to 7

# Collecting results
fifo_faults = [fifo(ref_string, frames) for frames in
frame_range]
lru_faults = [lru(ref_string, frames) for frames in
frame_range]
mru_faults = [mru(ref_string, frames) for frames in
frame_range]
optimal_faults = [optimal(ref_string, frames) for frames
in frame_range]

# Plotting
plt.figure(figsize=(10, 6))
plt.plot(frame_range, fifo_faults, marker='o',
label='FIFO')
plt.plot(frame_range, lru_faults, marker='o', label='LRU')
plt.plot(frame_range, mru_faults, marker='o', label='MRU')
plt.plot(frame_range, optimal_faults, marker='o',
label='Optimal')
plt.xlabel('Number of Frames')
plt.ylabel('Number of Page Faults')
plt.title('Page Replacement Algorithms: Frames vs Page
Faults')
plt.legend()
plt.grid(True)
plt.show()
```

Output :


Page Replacement Algorithms: Frames vs Page Faults
FIFO
LRU
MRU
Optimal
Number of Page Faults
20
18
16
14
12
10
8
6
Number of Frames
1
2
3
4
5
6
7