**OptimalPage**:

import java.util.Scanner;

import java.io.IOException;

public class optimalpg {

public static void main(String args[]) throws IOException {

Scanner sn = new Scanner(System.in);

int numberOfFrames, numberOfPages, flag1, flag2, flag3, i, j, k, pos = 0, max, hit = 0;

int faults = 0;

int temp[] = new int[10];

// Input number of frames

System.out.println("Enter number of Frames: ");

numberOfFrames = sn.nextInt();

int frame[] = new int[numberOfFrames];

// Input number of pages

System.out.println("Enter number of Pages: ");

numberOfPages = sn.nextInt();

int pages[] = new int[numberOfPages];

// Input the pages

System.out.println("Enter the pages: ");

for (i = 0; i < numberOfPages; i++) {

pages[i] = sn.nextInt();

}

// Initialize frame array with -1 (indicating empty)

for (i = 0; i < numberOfFrames; i++) {

frame[i] = -1;

}

// Process each page

for (i = 0; i < numberOfPages; i++) {

flag1 = flag2 = 0;

// Check if page is already in the frame (Page Hit)

for (j = 0; j < numberOfFrames; j++) {

if (frame[j] == pages[i]) {

flag1 = flag2 = 1;

hit++;

break;

}

}

// If page is not in frame (Page Fault)

if (flag1 == 0) {

// If there is an empty frame, add the page

for (j = 0; j < numberOfFrames; j++) {

if (frame[j] == -1) {

frame[j] = pages[i];

faults++;

flag2 = 1;

break;

}

}

}

// If no empty frame and no page hit, use Optimal Page Replacement

if (flag2 == 0) {

flag3 = 0;

// Check future uses of pages in the frames

for (j = 0; j < numberOfFrames; j++) {

temp[j] = -1;

for (k = i + 1; k < numberOfPages; k++) {

if (frame[j] == pages[k]) {

temp[j] = k;

break;

}

}

}

// If a page will not be used in the future, replace it

for (j = 0; j < numberOfFrames; j++) {

if (temp[j] == -1) {

pos = j;

flag3 = 1;

break;

}

}

// If all pages are used in the future, replace the one used farthest in the future

if (flag3 == 0) {

max = temp[0];

pos = 0;

for (j = 1; j < numberOfFrames; j++) {

if (temp[j] > max) {

max = temp[j];

pos = j;

}

}

}

// Replace the page in the frame

frame[pos] = pages[i];

faults++;

}

}

// Output the number of page faults and hits

System.out.println("\nTotal Page Faults: " + faults);

System.out.println("Total Page Hits: " + hit);

sn.close(); // Close the scanner

}

}