**Page replacement**

**1] FIFO**

import java.util.HashSet;

import java.util.LinkedList;

import java.util.Queue;

public class fifo {

static int pageFaults(int pages[], int n, int capacity) {

HashSet<Integer> s = new HashSet<>(capacity);

Queue<Integer> indexes = new LinkedList<>();

int page\_faults = 0;

for (int i = 0; i < n; i++)

{

if (s.size() < capacity)

{

if (!s.contains(pages[i]))

{

s.add(pages[i]);

page\_faults++;

indexes.add(pages[i]);

}

}

else

{

if (!s.contains(pages[i]))

{

int val = indexes.peek();

indexes.poll();

s.remove(val);

s.add(pages[i]);

indexes.add(pages[i]);

page\_faults++;

}

}

}

return page\_faults;

}

public static void main(String args[]) {

int pages[] = { 3, 5, 1, 0, 2, 3, 0, 3, 2, 4 };

int capacity = 4;

System.out.println("No of Page Faults are: " +

pageFaults(pages, pages.length,

capacity));

}

}

**No of Page Faults are: 7**

**2] LRU**

import java.io.\*;

public class lru {

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

BufferedReader obj = new BufferedReader(new InputStreamReader(System.in));

int f, page = 0, ch, pgf = 0, n, chn = 0;

boolean flag;

int pages[]; // pgf-page fault

System.out.println("1.LRU");

int pt = 0;

System.out.println("enter no. of frames: ");

f = Integer.parseInt(obj.readLine());

int frame[] = new int[f];

for (int i = 0; i < f; i++) {

frame[i] = -1;

}

System.out.println("enter the no of pages ");

n = Integer.parseInt(obj.readLine());

pages = new int[n];

System.out.println("enter the page no ");

for (int j = 0; j < n; j++)

pages[j] = Integer.parseInt(obj.readLine());

int pg = 0;

for (pg = 0; pg < n; pg++) {

page = pages[pg];

flag = true;

for (int j = 0; j < f; j++) {

if (page == frame[j]) {

flag = false;

break;

}

}

int temp, h = 3, i;

if (flag) {

if (frame[1] != -1 && frame[2] != -1 && frame[0] != -1) {

temp = pages[pg - 3];

if (temp == pages[pg - 2] || temp == pages[pg - 1])

temp = pages[pg - 4];

for (i = 0; i < f; i++)

if (temp == frame[i])

break;

frame[i] = pages[pg];

} else {

if (frame[0] == -1)

frame[0] = pages[pg];

else if (frame[1] == -1)

frame[1] = pages[pg];

else if (frame[2] == -1)

frame[2] = pages[pg];

}

System.out.print("frame :");

for (int j = 0; j < f; j++)

System.out.print(frame[j] + " ");

System.out.println();

pgf++;

} else {

System.out.print("frame :");

for (int j = 0; j < f; j++)

System.out.print(frame[j] + " ");

System.out.println();

}

} // for

System.out.println("Page fault:" + pgf);

}// main

}// class

**OUTPUT:**- enter no. of frames: 4

enter the no of pages 10

enter the page no

1

0

1

2

3

7

8

1

5

2

frame :1 -1 -1 -1

frame :1 0 -1 -1

frame :1 0 -1 -1

frame :1 0 2 -1

frame :1 3 2 -1

frame :7 3 2 -1

frame :7 3 8 -1

frame :7 1 8 -1

frame :5 1 8 -1

frame :5 1 2 -1

Page fault:9

**3] OPTIMAL**

import java.util.\*;

import java.io.\*;

public class Optimal {

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

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

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

int faults = 0;

int temp[] = new int[10];

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

numberOfFrames = Integer.parseInt(br.readLine());

int frame[] = new int[numberOfFrames];

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

numberOfPages = Integer.parseInt(br.readLine());

int pages[] = new int[numberOfPages];

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

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

pages[i] = Integer.parseInt(br.readLine());

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

frame[i] = -1;

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

flag1 = flag2 = 0;

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

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

flag1 = flag2 = 1;

break;

}

}

if (flag1 == 0) {

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

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

faults++;

frame[j] = pages[i];

flag2 = 1;

break;

}

}

}

if (flag2 == 0) {

flag3 = 0;

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;

}

}

}

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

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

pos = j;

flag3 = 1;

break;

}

}

if (flag3 == 0) {

max = temp[0];

pos = 0;

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

if (temp[j] > max) {

max = temp[j];

pos = j;

}

}

}

frame[pos] = pages[i];

faults++;

}

}

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

}

}

**Output:-** Enter number of Frames: 4

Enter number of Pages: 10

Enter the pages:

1

0

1

2

3

7

8

1

5

2

Total Page Faults:7