**Optimal Page Replacement**

#include <bits/stdc++.h>

using namespace std;

const int N=100005;

int n;

int frame\_size;

int pages[N];

void optimal\_page\_replacement()

{

vector<int> fr;

int page\_faults = 0;

int page\_hits = 0; // Add a variable to count page hits

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

{

int k;

for (k = 0; k < fr.size(); k++)

if (fr[k] == pages[i])

break;

if (k == fr.size())

{

if (fr.size() < frame\_size)

fr.push\_back(pages[i]);

else

{

int index = i + 1;

int res = -1, farthest = index;

for (int l = 0; l < fr.size(); l++)

{

int j;

for (j = index; j < n; j++)

{

if (fr[l] == pages[j])

{

if (j > farthest)

{

farthest = j;

res = l;

}

break;

}

}

if (j == n)

{

res = l;

break;

}

}

fr[res] = pages[i];

}

page\_faults++;

cout << "Reference to page " << pages[i] << " caused a page fault\n";

}

else

{

page\_hits++; // Increment the page hits count

cout << "Reference to page " << pages[i] << " did not cause a page fault\n";

}

}

cout << "\nTotal Page Hits: " << page\_hits << endl; // Output page hits count

cout << "Total Page Faults: " << page\_faults << endl;

}

int main()

{

cout << "Number of Frames: ";

cin >> frame\_size;

cout << "Page Reference Stream Length: ";

cin >> n;

cout << "Page Reference Stream:\n";

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

cin >> pages[i];

optimal\_page\_replacement();

return 0;

}

**Output:**

Number of Frames: 3

Page Reference Stream Length: 10

Page Reference Stream:

4 7 6 1 7 6 1 2 7 2

Reference to page 4 caused a page fault

Reference to page 7 caused a page fault

Reference to page 6 caused a page fault

Reference to page 1 caused a page fault

Reference to page 7 did not cause a page fault

Reference to page 6 did not cause a page fault

Reference to page 1 did not cause a page fault

Reference to page 2 caused a page fault

Reference to page 7 did not cause a page fault

Reference to page 2 did not cause a page fault

Total Page Hits: 5

Total Page Faults: 5