**Least recently used page replacement algorithm**

**As the name suggests LRU Page replacement Algorithm is based on the strategy that whenever a page fault occurs, the least recently used page will be the replaced with the new page. So the page not utilised for the longest time in the memory gets replaced. This is called as LRU paging.**

**A page fault occurs when a running program tries to access a page of memory. That is already not present in the main memory with that is RAM. if a page is already present in the memory RAM then it is called as Page hit. We use LRU page replacement algorithm only whenever a page fault occurred.**

**ALGORITHM**

Let capacity be the number of pages that

memory can hold. Let set be the current

set of pages in memory.

1- Start traversing the pages.

i) If set holds less pages than capacity.

a) Insert page into the set one by one until

the size of set reaches capacity or all

page requests are processed.

b) Simultaneously maintain the recent occurred

index of each page in a map called indexes.

c) Increment page fault

ii) Else

If current page is present in set, do nothing.

Else

a) Find the page in the set that was least

recently used. We find it using index array.

We basically need to replace the page with

minimum index.

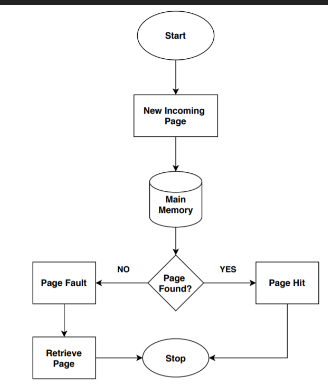
b) Replace the found page with current page.

c) Increment page faults.

d) Update index of current page.

**2. Return page faults.**

**FLOWCART**

****