**LRU Page Replacement Algorithm**

Let us say **s** that is the main memory's capacity to hold pages and pages is the list containing all pages currently present in the **main memory.**

1. Iterate through the referenced pages.

* **If the current page is already present in pages:**

**1.Remove the current page from pages.**

**2.Append the current page to the end of pages.**

**3.Increase page hits.**

* **Else:**

1. Increment page faults.
2. If pages contains less pages than its capacity **s**:

* Append current page into pages.

1. Else:

* Remove the first page from pages.
* Append the current page at the end of pages.

1. Return the number of pages hits and page faults.

Example :

Let there are 4 empty memory locations available.

Consider A reference string 5,3,4,2,5,7,6,7,2

**5 | F 3 | F 4 | F 2 | F 5 | F 7 | F 6 | F**

**( F -> Page fault)**

**7 | F 2 | F**