

2.

FIFO observations:

I implemented FIFO using a while loop which reads in all of the lines from accesses.txt and replaces pages according to whether they exist in the cache already. Each time the program reads another line, it checks to see if the input already exists in the cache. If it does, then nothing happens, but if the page does not already exist in the cache, the item at the current index is replaced with the new item.

LRU observation:

LRU depends on an extra variable, index, in the ref_page struct which keeps track of the age of each page in cache. If the page already exists in cache, then the age of all pages is increased and nothing happens. If the page does not already exist in the cache, then the oldest page is replaced with the new one.

Second Chance observations:

Second Chance depends on an extra variable, chance, in the ref_page struct which determines whether the page has had a second chance or not. If a page already exists in the cache and hasn't had a second chance, give it one. If it has already had a second chance, take it away. When a page no longer has a second chance and a new page is read in to the cache, replace the page with no second chance.

Sample page requests, page faults and hit rates

```
-----FIFO-----
10000 Total Page Requests
9916 Total Page Faults
0.008400 Hit Rate
-----End FIFO-----

-----LRU-----
10000 Total Page Requests
9915 Total Page Faults
0.008500 Hitrate
-----End LRU-----

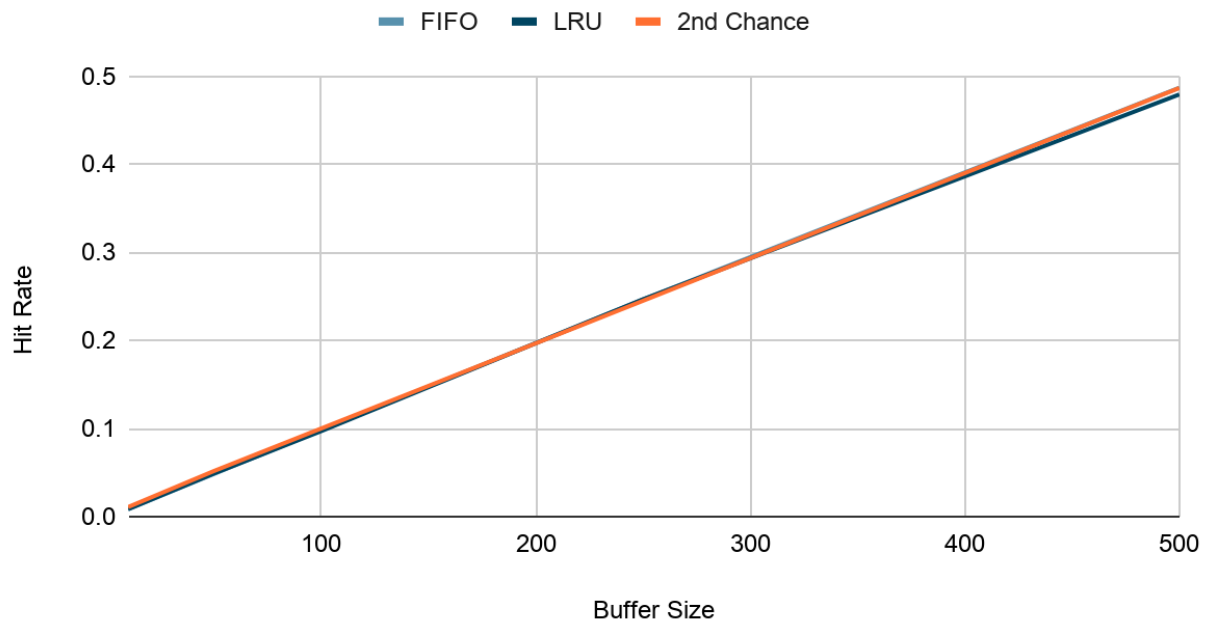
-----Second Chance-----
10000 Total Page Requests
9891 Total Page Faults
0.010900 Hit Rate
```

Table Comparing Hit Rate to Buffer Size

Buffer size	FIFO	LRU	2nd Chance	
10	0.0084	0.0085	0.0109	
50	0.0485	0.049	0.0518	
100	0.0982	0.0971	0.1001	
250	0.2466	0.2468	0.2451	
500	0.487	0.4794	0.4867	

3.

FIFO



As shown on the graph above, second chance has the highest hit rate for both a small and large buffer size. On the other hand, LRU performs the worst for both small and large buffer sizes but performs slightly better than FIFO and 2nd Chance at medium sized buffers. In most cases, 2nd Chance is the best algorithm to use because it performs much better at both of the extreme ends of buffer sizes while only performing slightly worse using medium sized buffers.