Results: Number of Faults

N = 16

	FIFO	LRU	CLOCK
Workload Original	250440	250440	250440
Workload 1	97657	97657	97657
Workload 2	50000	50000	50000

N = 32

	FIFO	LRU	CLOCK
Workload Original	250424	250424	250424
Workload 1	97657	97657	97657
Workload 2	50000	50000	50000

N = 1024

	FIFO	LRU	CLOCK
Workload Original	249504	249508	249508
Workload 1	97657	97657	97657
Workload 2	50000	50000	50000

Access Patterns

For Workload Original:

The access pattern is to access 2 consecutive values of a random page.

For Workload 1:

The access pattern is to access all entries consecutively and wrap around to the table's start if we go over the MAX_SQRTS limit.

For Workload 2:

The access pattern is similar to Workload 1 but we access each 1000th entry instead of consecutive entries.

Algorithm performance

Looking at all three patterns, we see that for Workload 1 and 2, we don't access a previous page until after we have wrapped around the whole table. Each page contains 512 entries, and the whole table contains 2^18 pages. So on Workload 1 and 2, FIFO, LRU and CLOCK algorithms will perform the same as our N is very small, and we would always evict the page before we come back to it.

But for Workload Original, our access pattern on pages is random. We still see similar page faults on all three algorithms. That is because the number of pages is very large, and the probability of accessing the same page is very low. If the probability is low, the page would have already been evicted out before it's accessed. As we increase N, the algorithms will start showing lesser faults, but N is still not large enough to see significant improvement.

Considering the above results, the FIFO algorithm is the best as it has the least overhead in calculating the victim page and the memory overhead of keeping the page metadata. Second is

LRU and last is CLOCK due to their increasing overheads of calculation and page metadata overhead.