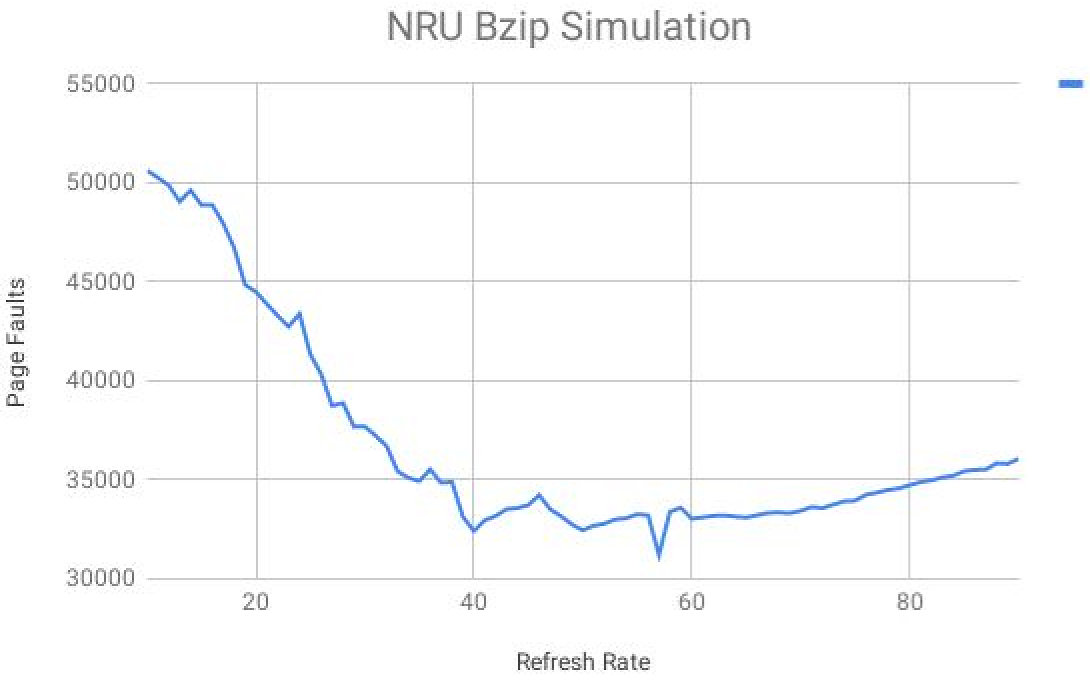
Michael Korst

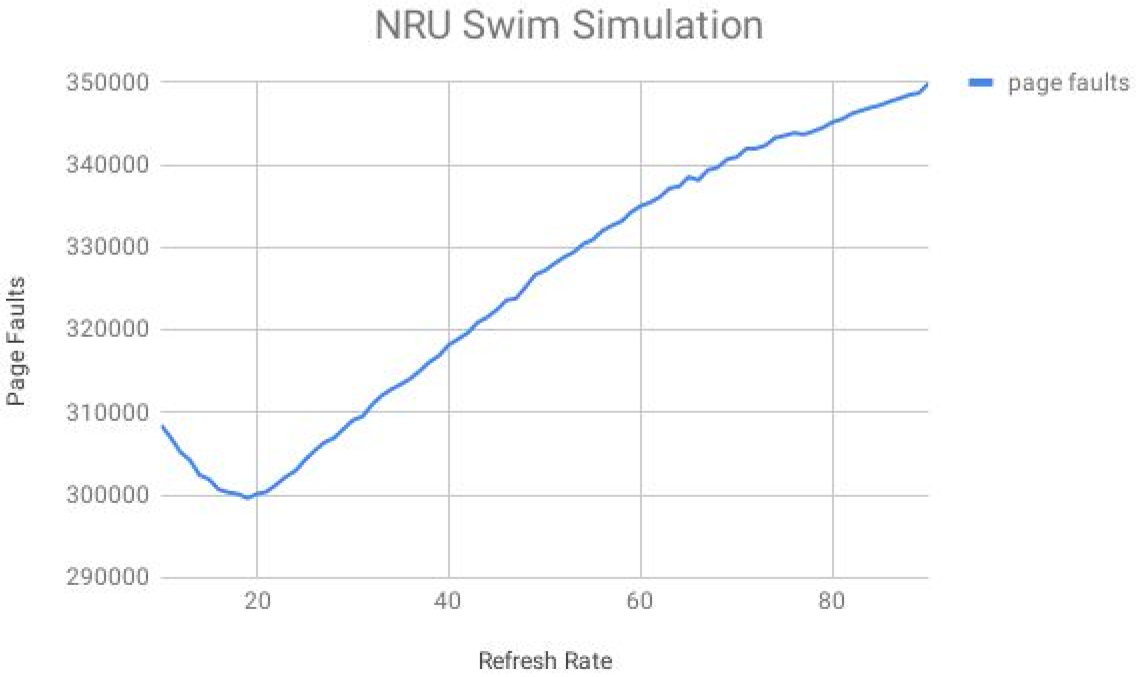
CS 1550

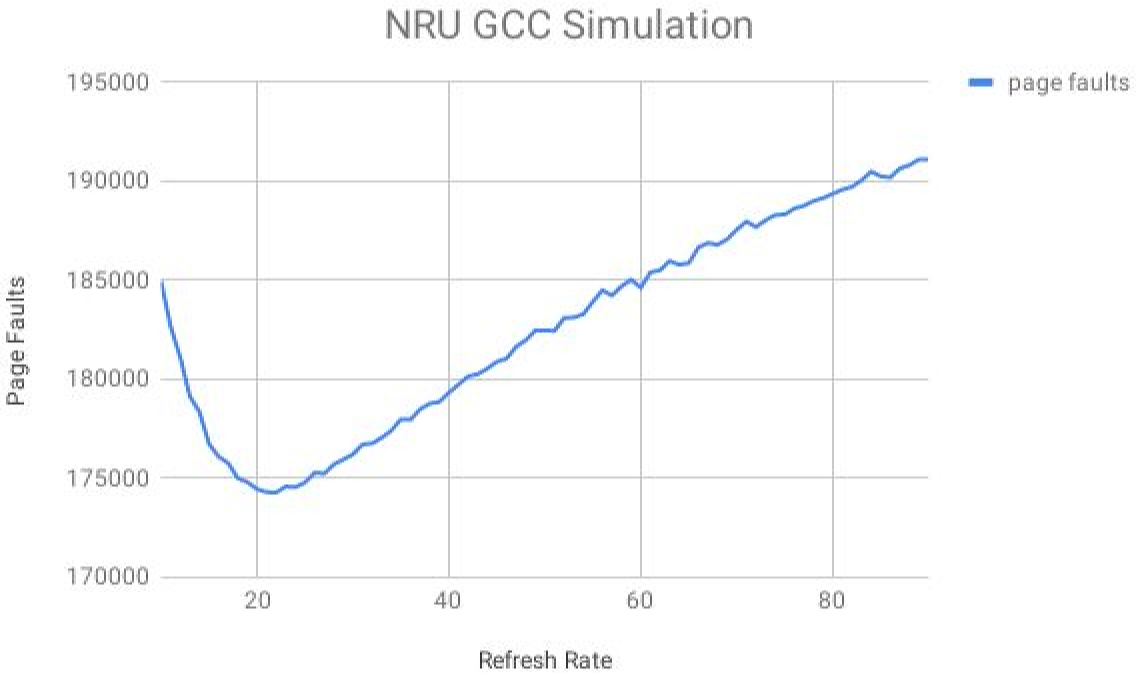
12 November 2018

Project 3 Write-Up

1) Determining optimal value for NRU refresh parameter, ranging refresh rate from 10 to 90 (for each of the 3 trace files):







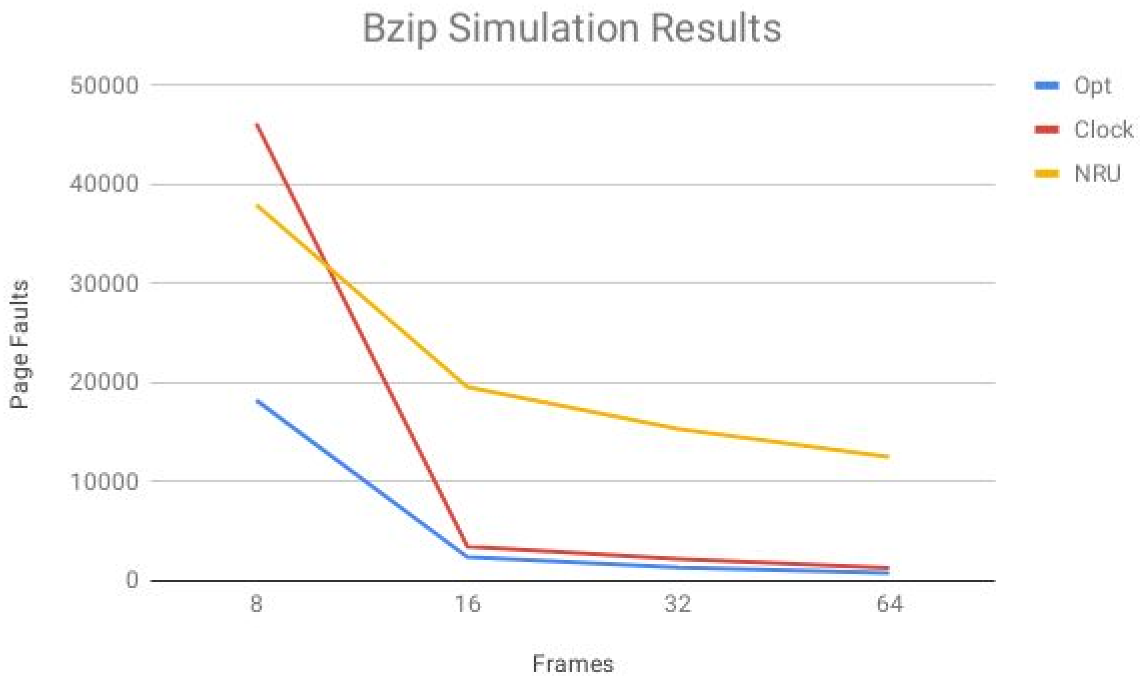
Based on iterated simulations done on the 3 trace files, the minimum of this range for page faults seems to be somewhere 20 for the last 2 and 45-50 for the 1st bzip sim. Based on this, I think 30 would be a reasonable refresh rate to pick as “optimal”. (Note: 8 frames were used for all of these simulations).

2) Below is a comparison of page fault rates in all 4 algorithms for frame sizes of 8, 16, 32 and 64. For NRU the previously determined refresh of 30 is used.

Bzip Sim Results:

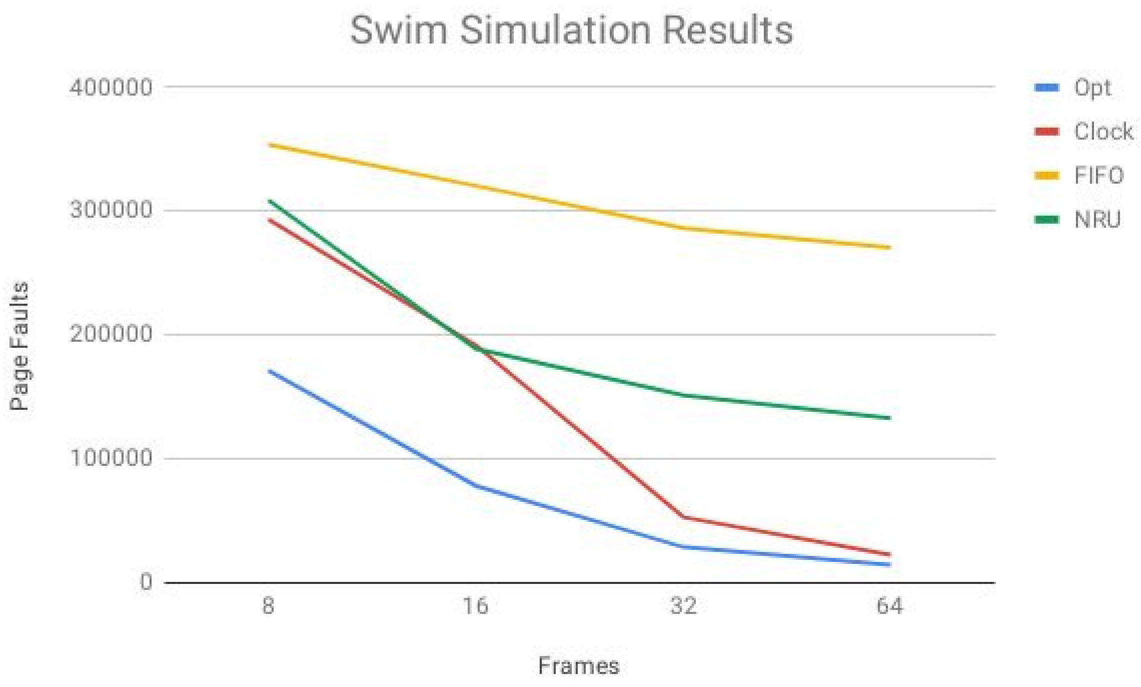
Opt		Clock		FIFO		NRU	
Frames	Page Faults	Frames	Page Faults	Frames	Page Faults	Frames	Page Faults
8	18251	8	46164	8	626041	8	37953
16	2427	16	3468	16	625211	16	19589
32	1330	32	2203	32	616985	32	15340
64	821	64	1318	64	550293	64	12527

Excluding the FIFO results from the graph (as they are too large), here is a graph of Opt, Clock and NRU for Bzip.



Swim Sim Results:

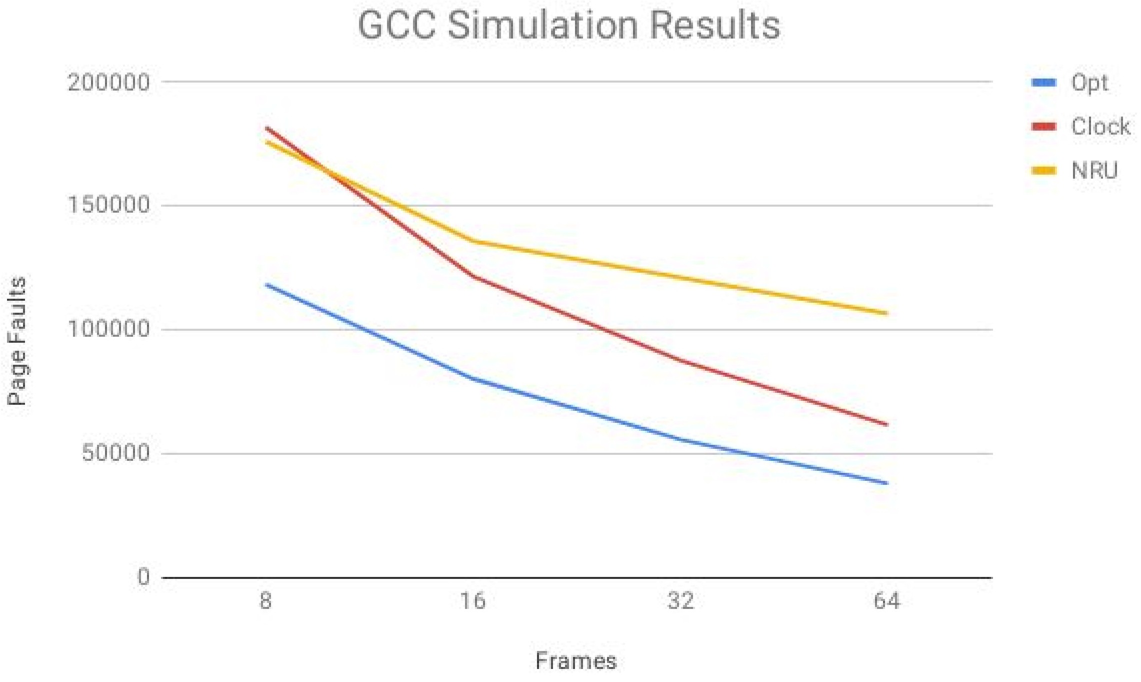
OPT		Clock		FIFO		NRU	
Frames	Page Faults	Frames	Page Faults	Frames	Page Faults	Frames	Page Faults
8	171244	8	293519	8	353958	8	309038
16	78312	16	191848	16	320459	16	188867
32	28826	32	53025	32	286282	32	151360
64	14289	64	22611	64	270479	64	132947



GCC Sim Results:

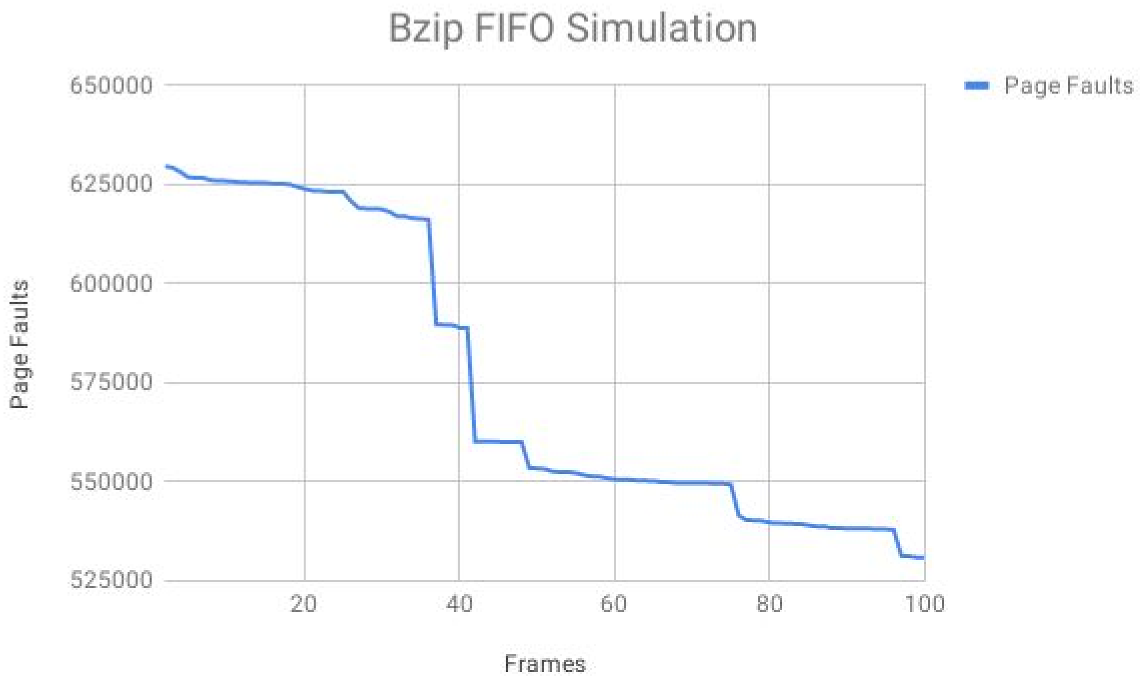
OPT		Clock		FIFO		NRU	
Frames	Page Faults	Frames	Page Faults	Frames	Page Faults	Frames	Page Faults
8	118480	8	181856	8	715721	8	176099
16	80307	16	121682	16	702468	16	135967
32	55802	32	87686	32	634404	32	121222
64	38050	64	61640	64	447121	64	106691

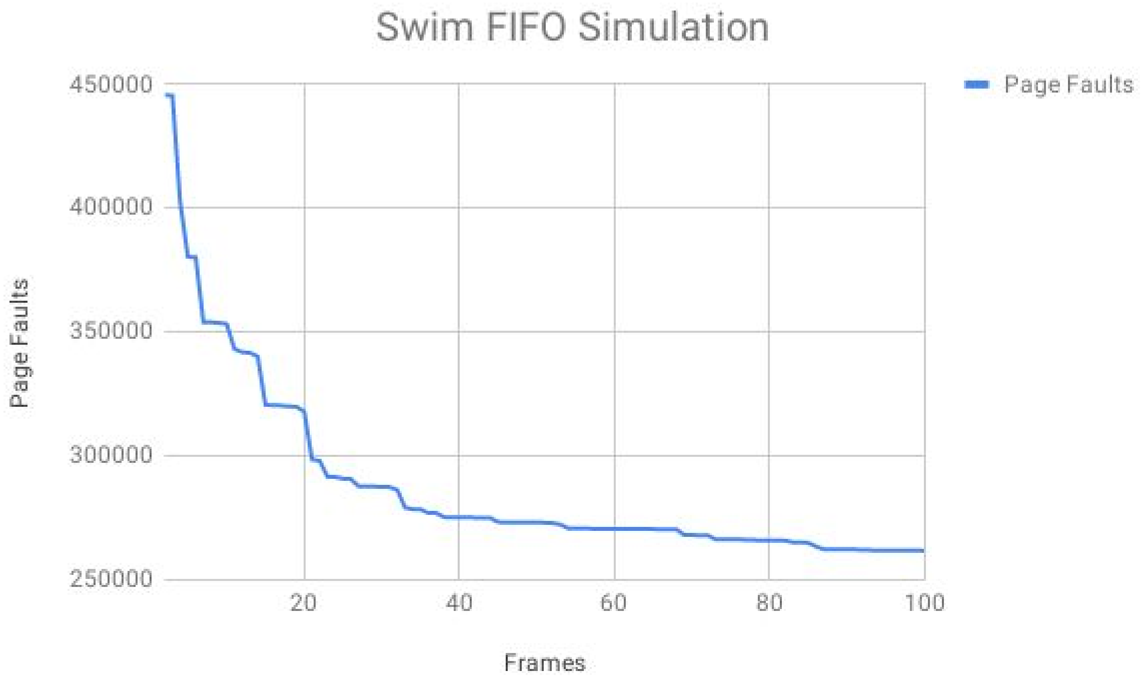
Here is the graph (once again excluding FIFO due to its massive difference in range):

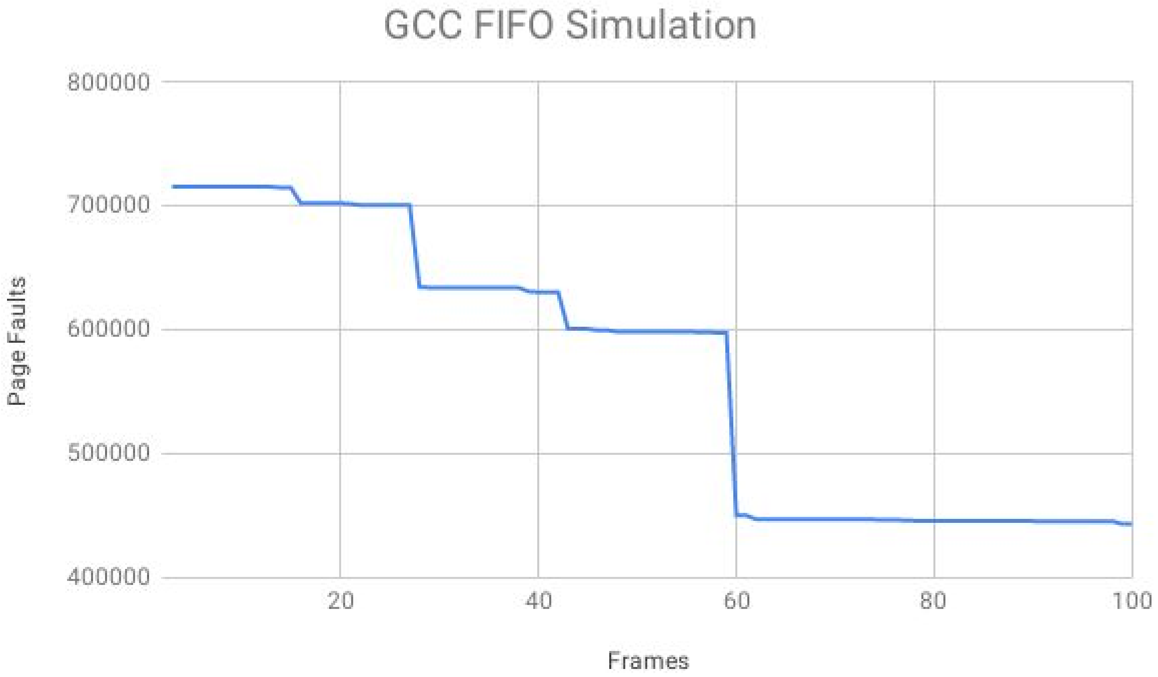


Conclusion: Based on the simulation numbers and charts, it appears that Clock is the closest algorithm to optimal. It certainly seems to largely follow the trends of Opt as it responds to an increase in frames. FIFO is, as expected, the least optimal algorithm by far. It takes no priority into consideration and is thus not suitable for use in an actual OS. The only contender to the best algorithm other than Clock is NRU. As we can see, based on these simulations, Clock appears to have the edge on NRU. However, NRU is heavily dependent on an optimized refresh rate. In a real system, this could be determined by many different factors and varies. So, because of this, if optimized properly, NRU might be able to be the best algorithm. However, based on these observations from the trace files, Clock is the algorithm with the least page faults, the closest pattern to Opt, and is thus the optimal algorithm for a real operating system.

3) Below is a chart showing the page fault rate for FIFO on each of the three traces, varying the frames from 2 to 100:


Bzip FIFO Simulation
Page Faults
650000
625000
600000
575000
550000
525000
20
40
60
80
100
Frames
Page Faults


Swim FIFO Simulation
Page Faults
450000
400000
350000
300000
250000
20
40
60
80
100
Frames
Page Faults



Based on the iterations shown above, I did not find any instances of Belady's anomaly. As we can see, as frames increase, the page faults appear to decrease, as we would expect. There is no instance of the anomaly where we would see the faults increasing with an increase in frames. However, it's still important to note just how sub-optimal FIFO is. The algorithm doesn't appear to become much more efficient even with vast amounts of frames. This is why we generally avoid FIFO in Operating System design.