**Belady’s Anomaly and its relation to the law of diminishing returns**

I’ve noticed that with Belady’s anomaly that without a sufficient main memory size to hit rate\*, the cost to store the main memory can be more expensive than the savings of retrieving from main memory because you’re replacing main memory faster than you’re getting a return on using it.

**\* For Let Hit Rate be defined by the (Number of Requests-Page Faults)/the number of misses.**

Hypothesis: Due to this observation, the change in hit rate in the FIFO Algorithm follows the law of diminishing returns due to Belady’s anomaly. This, in turn, supports the definition of the Operating System being a resource manager and thus simulating the definition of a closed economy.

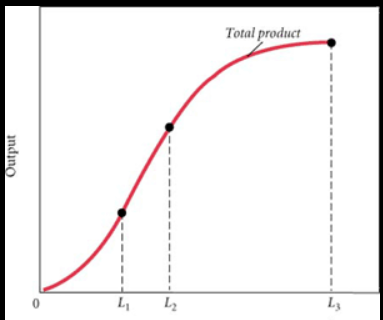
However, since the Hit Rate will never be negative ( since only the table size will be increasing), the change in hit rate will stop at around 0.

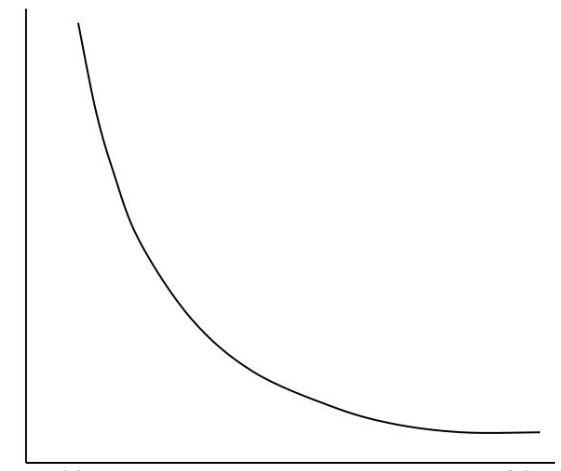
According to Wikipedia In economics, the law of diminishing returns is the decrease in the marginal (incremental) output of a production process as the amount of a single factor of production is incrementally increased while the amounts of all other factors of production stay constant. The law of diminishing also returns states that in all productive processes adding more of one factor of production while holding all others constant will at some point yield lower incremental per-unit returns.

Let the first graph above be labor usage to output, and the second graph is the derivative of the first.

X Axis for both Graphs are labor usage

And Y Axis for both graphs are for output.





In the case of Belady’s anomaly, the production output would become the hit rate and its derivative would become the marginal output. The labor usage would be a memory.

Because Belady’s anomaly is unbounded, according to Peter Fornait and Antal Ivanyi of Eötvös Loránd University, there will only be 1 turning point of the relative max.

(pdfs.semanticscholar.org/22cd/e61e588918eded2866276080e36c00a138a5.pdf)

Mathematically since this would make sense because if we are increasing just the table size, the limit would

Proof:: Limit = d hitrate /dtablesize= 0

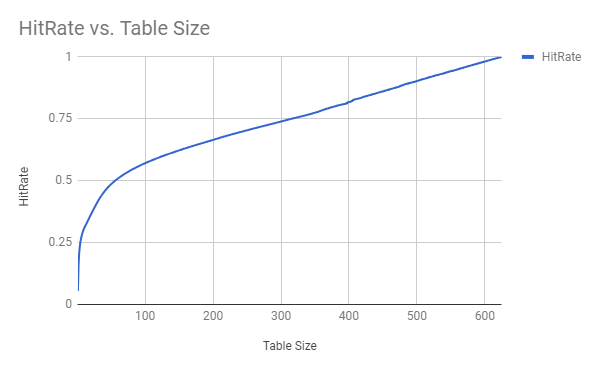
Table size\_> infinity

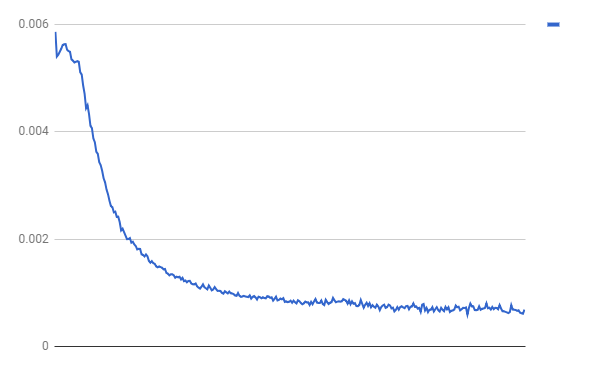
Where D is delta\*

In this case Change in hit rate will be decreasing over time but the change in table size will remain constant.

To test my hypothesis, I implemented the FIFO algorithm from assignment 4, but I incremented the table size from 1 to 650 for maximum accuracy. Then, I graphed the rate of change by graphing the slope.

The results graphs are very similar to hypothetical economic graphs above. The bumps are reasonable since the memory increments are by 1.





The graph above is the derivative of hit rate.

The Y axis is the change in hit rate and the x axis is the table size

Clearly, the return rate has been decreasing over time, thus simulating the law of diminishing marginal return graphs shown in page 2.

Around table size 400, we can see that the change in hit rate has been leveled, so the programmer should be aware that if he uses a nonstack algorithm, that there a point where the increase in hit rate is not worth the tradeoff in memory.

Thus, When a programmer implements FIFO or any nonstack algorithm, in general, he should be aware of when the hit rate decrements as that will be replacing main memory faster than you’re getting a return on using it. Since this follows the standard economic law of diminishing returns