REPORT OF CS3230 PJ1 TASKB

Efficiency Analysis of Replacement Policies

I. To compute each entry in C, we need $p + p + 1 = 2p + 1$ accesses to A ,B and C. So the total number is $(2p+1)\times(nm) = O(npm)$, which equals to the order of cache miss when cache size is small.

II. I test these four policies with some data, compare the trends and try to explain it.

A. FIFO

It's not bad but much worse than OPT yet. Many victims from B will be used again actually but we remove them from cache since they are early in.

B. LIFO

It has high miss rate. Because it keeps a lot of useless entries of A in bottom but often removes useful entries of B in top.

C. LRU

It has even a bit higher miss rate than LIFO. The row of A in current computation of corresponding row of C is always removed to the younger end. So this row occupies cache room in a long time even though it's not useful any more.

D. OPT

It has the lowest (far lower than the second best one) miss rate because it exploits the knowledge of future. Also that's why it's called so.