

Instructions

1. You are encouraged to work in groups of two. If you cannot find a partner, you can work alone.
2. Please submit one copy of the assignment; if you are working with a partner, both names should appear on the assignment.
3. For **Part A** of the assignment, you must submit an electronic copy of your Java application via WebCT (by the time of the lecture on the due date of the assignment).

Part A

Write a program which does the following experiment: on input m, n, k it creates a random list of m requests, where each request is from among $[n] = \{1, 2, \dots, n\}$ pages of memory, and it counts the number of misses on a cache of size $i \leq k$ of each of the following page replacement algorithms: LRU, CLOCK, FIFO, LIFO, LFU, LFD¹, and plots the result.

In other words, for each $0 < i \leq k$ it draws a point in position $(i, \text{misses}_{\mathcal{A}}(i))$, the number of misses on a cache of size i on algorithm \mathcal{A} . (For a better visual effect, plot the outcome of each algorithm in a different color, and provide a legend.)

¹See page 43 of the notes for the description of these algorithms.

Part B

1. For a general i , provide a sequence of page requests that illustrates Belady's Anomaly incurred by FIFO on cache sizes i and $i + 1$. In your analysis, assume that the cache is initially empty.
2. Show that FWF *is* a marking algorithm, and that FIFO *is not* a marking algorithm.
3. We say that a page replacement algorithm ALG is *conservative* if it satisfies the following condition: On any consecutive input subsequence containing k or fewer distinct page requests, ALG will incur k or fewer page faults. Prove that LRU and FIFO are conservative, but FWF is not.