Concordia University
Dept. of Computer Science and Software Engineering
COMP 691 – Online Algorithms and Competitive Analysis
Winter 2024

$\mathcal{P}$roject

1 Phase 2: extension of code base

Deadline: submission on Moodle by 23:55PM on April 1, 2024

What to submit: a single .zip file with source code + readme.txt

Weight: 3% of the overall course grade

Description of the phase: prior to beginning this phase, you should check your grade and comments for phase 1. If there were any issues with your phase 1 code, you should fix those issues. In addition, you should expand the code base you started in phase 1 by implementing the following functions:

1. $LRU(k, seq)$. The input is parameter k - cache size, seq is the sequence of page requests $p_1, \ldots, p_n$.

2. $combinedAlg(k, seq, hseq, thr)$. The input is parameter k - cache size, seq is the sequence of page requests $p_1, \ldots, p_n$, $hseq$ is the sequence of predictions $\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_n$, and thr is a number between 0 and 1 that indicates a threshold parameter. The output is the number of page faults incurred by *Combined* algorithm, which works as follows. The *Combined* algorithm keeps track of f_1 - number of page faults incurred on the input so far by LRU, and f_2 - the number of page faults incurred on the input so far by *blindOracle*. Thus, the algorithm pretends to run LRU and *blindOracle* side-by-side.

   Here is how *Combined* algorithm makes online decisions: it starts out by maintaining same cache contents and doing same decisions as LRU. If in a step i it is determined that $f_1 > (1 + thr)f_2$ (that is LRU has made too many mistakes compared to *blindOracle*), then the algorithm switches to *blindOracle* cache contents and decisions. This switch is achieved by clearing the cache, populating its contents with cache contents of *blindOracle* at the time of the switch, and continuing to use *blindOracle* decisions until $f_2 > (1 + thr)f_1$. When $f_2 > (1 + thr)f_1$ the algorithm switches from *blindOracle* to LRU, and so on. Thus, the combined algorithm keeps switching between LRU and *blindOracle* depending on whichever one has made fewer mistakes so far, up to the multiplicative threshold parameter. Note that each switch adds k page faults to the total count of page faults incurred by the *Combined* algorithm. Note that you should keep careful track of number of page faults of LRU, *blindOracle*, and *Combined*. You should clearly understand the differences between the number of page faults of each, and your function should return the total number of page faults incurred by *Combined* algorithm.

3. $test1(), test2(), test3(), \ldots$. These are functions that you can use to test the functionality of the above functions.

4. $main()$. This function runs a bunch of test functions, and reports if everything is working as expected.

Note: the same comments on code, comments, documentation, readme file as for phase 1 of the project apply.

2 Phase 3: experiments and report

Deadline: submission on Moodle by 23:55PM on April 15, 2024

What to submit: a single .zip file with source code + readme.txt + report.pdf

Weight: 4% of the overall course grade

Description of the phase: TBA