Institut for Matematik og Datalogi
Syddansk Universitet

February 26, 2019
JFB

Online Algorithms – F19 – Assignment 1

Assignment due Tuesday, March 19, 10:15

This is the first of three sets of problems (assignments) which together with the oral exam in June constitute the exam in DM860. This first set of problems may be solved in groups of up to three.

The assignment is due at 10:15 on Tuesday, March 19. You may write this either in Danish or English. If your assignment is late, it will not be accepted. Turn in the assignment through the SDU Assignment system in Blackboard, and remember to keep your receipt. Turn in one PDF file per group.

Cheating on this assignment is viewed as cheating on an exam. Do not talk with anyone outside of your group (or Joan Boyar) about the assignment, and do not show your solutions to anyone outside your group. If you have questions about the assignment, come to Joan Boyar.

Explain all of your answers.

1. Show how to use part of the proof of Theorem 1.2 in the text book to get a stronger result for MTF in the static list model than the $2 - \frac{1}{\ell}$ shown above Exercise 1.3 in the textbook.

2. On problem 5 on the first lecture note for this course, MTF2 is defined, and a family of sequences proving a lower bound of $\frac{5}{2}$ on MTF2's competitive ratio is given. The last part of that problem asks how to change that family of sequences so that it also demonstrates the same lower bound for MTFE, the algorithm which moves to the front on every even access, instead of every odd. We did not do that in class. How do you do this?

3. Give a family of sequences which gives an asymptotic lower bound on TIMESTAMP's competitive ratio (in the full cost model) of $\frac{2\ell}{\ell+1}$, where ℓ is the length of the list. Note, this should hold for any $\ell \geq 1$.

4. Give pseudocode (a more detailed algorithm than in the textbook, indicating how you determine if there have been two or more accesses to an element in between two accesses to another) for the TIMESTAMP algorithm. Do you need one, two, or more timestamps per element (try for a smaller number, rather than a larger one)? How many bits are needed for representing a time stamp (how large are the numbers that you would keep)?

5. Do Exercise 3.9 in the textbook.

6. Fix some k and N and find a request sequence where FIFO does better than LRU and another where LRU does better than FIFO.

7. Suppose one considered a Modified Competitive Ratio, R', such that each algorithm A, for minimization problem P, is compared to an "optimal" offline algorithm OPT_A, which is optimal among all algorithms such that for any sequence σ and any prefix σ' of σ, $\mathrm{OPT}_A(\sigma') \leq A(\sigma')$. Note that there can be a different OPT for each algorithm, so the standard OPT used in competitive analysis cannot always be used. For example, consider the bin packing problem, the algorithm First-Fit, and the request sequence $\langle \frac{1}{3}, \frac{1}{2}, \frac{1}{3}, \frac{1}{2}, \frac{1}{3} \rangle$. $\mathrm{OPT}_{\mathrm{First-Fit}}$ has to put the first two requests, $\frac{1}{3}$ and $\frac{1}{2}$ in the same bin, since otherwise on the prefix $\langle \frac{1}{3}, \frac{1}{2} \rangle$, it does worse than First-Fit. (See section 12.5 in the textbook for definitions and results related to bin packing and First-Fit.)

   As with the original competitive ratio, for an algorithm A to have a Modified Competitive Ratio of c, there must exist a constant b such that for all sequences σ, $A(\sigma) \leq c\mathrm{OPT}_A(\sigma) + b$.

   (a) Find a sequence for bin packing where First-Fit and $\mathrm{OPT}_{\mathrm{First-Fit}}$ are not identical.

   (b) Prove that $\frac{17}{10}$ is also an upper bound on the Modified Competitive Ratio of First-Fit.

   (c) Show that for the paging problem and any algorithm A for the paging problem, letting OPT_A be the optimal offline algorithm, LFD, satisfies the above definition. Use this result to determine the Modified Competitive Ratio of LRU.

8. Consider a weighted paging problem where each page has a weight (cost) for bringing it into cache. Suppose ALG always evicts the lowest

weight page (because it would cost least to bring in again) on a page fault. Prove that ALG is not competitive.