Comp 7720 - Online Algorithms

Assignment 2: Compression, Splay Trees, Caching, and $k$-server

Shahin Kamalli
University of Manitoba - Fall 2019

Due: October 14th at 11:59 pm

“Some people talk to animals. Not many listen though. That’s the problem ...”
Winnie-the-Pooh (A. A. MILNE)

Happy October 4th, World Animal Day

Please pay attention to the followings when preparing/submitting your assignment:

- All problems are written problems. There are six problems with a total of 55 marks. The last problem is a bonus problem (your assignment will be marked out of 45). The bonus problem is harder than the rest and will be marked in a different way. Approach it only after you finished other questions.

- If you have any question related to the assignment, you are encouraged to post it on Piazza. Note that you can submit anonymously. Also, instead of emailing me, you can always write a private note in Piazza. It is likely that I drop hints when a question is posted publicly on Piazza (because all students can benefit from it). It is not the case when you ask questions in emails or during office hours.

- You are welcome to discuss the problems with your friends (or enemies). But you should write your answers individually. You might be interviewed about your answers. Be careful not to accidentally copy.

- Submit your answers electronically using Crowdmark. You should answers for different questions separately.
Problem 1  Compression [10 marks]

a) Apply the Burrows-Wheeler transform on the following string; show your work and the output.

HONEYPONY$

Assume $ precedes all characters when you sort rotations.

b) Assume an initial list $ → A → B → E → H → N → O → P → Y$, i.e., initially $ is at index 0, A is at index 1, etc.

Assume we use Move-To-Front on the above list to encode the outcome of the BWT transform from part (a). Show what numbers will be encoded (you need to show how the list is updated).

c) Assume an initial list $ → A → B → C → D$, i.e., initially $ is at index 0, A is at index 1, etc. Assume we use Timestamp on the above list to encode DB$AABCD. Show what numbers will be encoded (you need to show how the list is updated and in particular its last state).

d) Assume an initial list $ → A → B → C → D$. A compressing scheme that uses Move-To-Front has encoded the following numbers for a text T. Show what the actual text is. The numbers are 2 2 0 3 3.

Problem 2  Splay Trees [10 marks]

a) Apply the splay operation on the following splay tree when there is a request to node ‘8’. Show your steps.

b) Consider a splay tree on m nodes. Assume the next access is to a node that is selected uniformly at random. What is the expected number of the children of the root after the access? (you should provide a number as a function of m).

c) Prove or disprove the following statement: “after a splay operation, the old root is always at depth 1 or 2 (i.e., it is at distance 1 or 2 of the new root)”.

Problem 3  Paging & Resource Augmentation [5 marks]

Sometimes when we analyse online algorithms, we reduce the power of Opt to be more fair to the online algorithm; it is called resource augmentation. For example, for the paging algorithm, instead of comparing an online algorithm with an optimal algorithm which has the same cache-size, we assume the size of the cache of Opt is smaller than that of algorithm. Clearly, when comparing with a weaker optimal algorithm, the competitive ratio of algorithms is expected to improve.

Assume the cache of a marking algorithm A has size k and the cache of Opt has size k/5. Provide an upper bound for the competitive ratio of LRU under this setting.
**Problem 4  \( k \)-Server Lower Bound [10 marks]**

Consider the \( k \)-server under resource augmentation setting (see Question 3) where the offline algorithm has \( k - 1 \) servers while the online algorithm has \( k \) servers. Adapt the lower bound that we saw in the class to show a lower bound for the competitive ratio of any deterministic algorithm for this setting.

**Problem 5 Double-Coverage-Algorithm [10 marks]**

(a) Consider a different analysis of the double coverage algorithm in which the potential is defined as \( 3(P + Q) \), where \( P \) and \( Q \) are defined as before. Follow the steps of the algorithm and indicate whether we can achieve an upper bound for the competitive ratio of the algorithm with this way of defining the potential (if the answer is yes, you should show what that competitive ratio is).

(b) Consider a different analysis of the double coverage algorithm in which the potential is defined as \( P + 3Q \), where \( P \) and \( Q \) are defined as before. Follow the steps of the algorithm and indicate whether we can achieve an upper bound for the competitive ratio of the algorithm with this way of defining the potential (if the answer is yes, you should show what that competitive ratio is).

**Problem 6  [Bonus] Double-Coverage-Algorithm [10 marks]**

Consider a variant of the double coverage algorithm for paths that works as follows. Upon a request to a vertex \( x \), if there is only one server on its two sides, that server is moved to serve the request (as before). If there are two servers on the two sides of the request, both move towards the request (as before). The left server, however, moves at twice the speed of the right server. In the below picture for example, if there is a request to 5 at time \( t \), the left server \( s_1 \) gets there before \( s_2 \), despite being further initially.

Follow the same steps as we took in the analysis of the double coverage algorithm to provide an upper bound for this variant of the algorithm.