Institut for Matematik og Datalogi
Syddansk Universitet

May 6, 2013
JFB

DM508 – Algorithms and Complexity – 2013
Lecture 9

Lecture, May 6

We finished Fibonacci heaps (starting with the Extract-Min operation) and began on string matching from chapter 32, covering the naive algorithm and (most of) the Rabin-Karp algorithm.

Lecture, May 13

We will finish string matching from chapter 32.

Lecture, May 15

Class may be cancelled if we have finished string matching. Otherwise, we will finish string matching.

Problems to be discussed on May 21

Do problems:

1. 32.4-1, 32.4-3, 32.4-6, 32.4-7.
2. 32.4-4 and 32.4-5.

Assignment due Wednesday, May 22, 14:15

Note that this is part of your exam project, so it must be approved in order for you to take the exam in June, and you may not work with or get help from others not in your group (though you may talk with Marie Christ or myself). No part of your work may be taken from another source. You may work in groups of two or three. You may write your solutions in English or Danish, but write very neatly if you do it by hand. Submit the assignment via Blackboard's "SDU Assignment" as one PDF file. Turn in one assignment per group.

1. A queue can be implemented using stacks. Recall that with a queue you want to ENQUEUE and DEQUEUE, so that you insert in the opposite end from where you delete items (in a first-in-first-out manner). If you have the standard PUSH and POP operations from a stack, you can implement the queue operations, using two stacks, StackA and StackB.

   To implement ENQUEUE(Q,X), execute PUSH(StackA,X).

   To implement DEQUEUE(Q,X), if StackB is empty, first execute

   WHILE StackA IS NOT EMPTY, POP(StackA,Y), PUSH(StackB,Y), END WHILE.

   In both cases (whether or not StackB was empty), end by executing POP(StackB).

   Note that when the while loop is executed, it reverses the order of StackA. Convince yourself that this works correctly. In what follows, "time" is used to mean the total number of push and pop operations.

   (a) Prove that with this implementation, a sequence of n queue operations can result in one dequeue operation taking time $\Omega(n)$.

   (b) Use amortized analysis with a potential funciton to prove that a sequence of enqueue and dequeue operations takes $O(n)$ time in the worst case. In your proof, make it clear what the potential function is and what the amortized running time of each of the two operations is.

2. A Fibonacci heap is not designed to have short paths from a root to a leaf. Show a sequence of operations which would lead to a path of length 4 and only 4 nodes in the tree. Explain how to generalize this to any n. How many operations are done on the heap (as a function of n)?

3. What is the prefix function computed by the KMP algorithm for the string P = abbabbabcabbabbc.