COMP20003 Algorithms and Data Structures

Worksheet 5

[week starting 21 of August]

Second (Spring) Semester 2016

Overview

The workshop for Week 5 will start with a tutorial on Stacks and Queues.

Tutorial Questions

Question 5.1 A `stack` is an abstract data structure that behaves much like a stack of books on a desk: The newest additions go on top, and the newest additions are also the first to come off the stack.

Show:

1. How you could implement a stack using an array as the underlying data structure. Think about how you would deal with the situation where the stack is full.
2. How you would implement the stack using a linked list as the underlying data structure.

Question 5.2 A `queue` is an abstract data structure that behaves much like a queue in a grocery shop: The newest additions to the queue go at the end (tail), and items are taken out of the queue and processed from the front (head).

Show:

1. How you could implement a queue using an array as the underlying data structure. Think about how you would deal with the situation where the queue is full. Discuss whether a `circular array` can be used.
2. How you would implement a queue using a linked list as the underlying data structure.

Discuss the pros and cons of the linked list and array implementations.

Programming exercises

Programming 5.1 In the tutorial section we discussed `stacks` and `queues` implemented using arrays or linked lists as the underlying data structure. In this exercise, you are asked to implement a stack based on a linked list. This exercise should give you practice in using `malloc()` to make nodes, filling up the fields within nodes, and abstracting implementation details into functions.

You should have at least the operations (functions) `makeStack, push, pop`. Having an operation to check whether the stack is `empty` is also a good idea. Separate each stack operation into a different function! The functions should take a pointer to the stack (list) as an argument, so that multiple queues can operate within a program using the same functions, and possibly other arguments, such as the item to be pushed.

For an extra challenge, consider how you would free all the memory used by the stack once your program is finished with it.

August 19, 2016