CS2040: Data Structures and Algorithms

Tutorial Problems for Week 5: Lists and ADTs

Hints: Stuck? Don't abandon the problem completely! See if you can make some progress towards the solution with the hints here.

Problem 1. True or False?

This question is for you to check your understanding of the data structures. Therefore, no hints here! We will discuss the problems during tutorial.

Problem 2. Circular Linked List

Hint: First, note that the index of the node given may be larger than the size of the list, and therefore the index wraps around. Is there any way to remove the wrap around for the indexing?

Next, try to draw a circular linked list containing 5 nodes out on a piece of paper. Recall that every node in a linked list stores a pointer/reference to the next node in the linked list. Let's say we swap the 3rd and 4th node in the circular linked list. Which nodes contain pointers that are affected by the swap? You need to ensure that your algorithm correctly modifies all the pointers affected by the swap. After you have worked out an algorithm, try to simulate the swap on the linked list that you have drawn step by step and see if it works.

Finally, what are the special cases that we need to handle? For example, if the linked list contains only one node, then we actually do not need to perform any swapping because we can only swap the node with itself, and after the swap is performed, the linked list is exactly the same as the one before the swap. Can you think of any other special cases? You need to ensure that your algorithm handles all the special cases.

Problem 3. Waiting Queue

Hint: In this problem, we need to implement a queue with a delete operation. This means that our queue should still work like a standard FIFO queue, but we should be able to delete elements in the middle of the queue as well.

Since we are implementing the queue using an array, deleting an element from the queue is equivalent to deleting an element from an array: we can remove the element, and shift all the elements behind it forward to 'cover up the gap'. This is clearly slow: for a queue with n elements, this method of deletion may work in $O(n)$ time. Think about this: since deleting is costly, do we need to delete the element immediately? Is there some way to allow us to 'delay' the deletion in a way such that we can perform the deletion later, but a lot faster?

Problem 4. Stack Application – Expression Evaluation

Hint: This problem is quite similar to the ‘Evaluating Arithmetic Expression’ problem discussed in lecture. Look at how the example was done in lecture, and try to apply similar ideas. As an additional hint, you may need two stacks for this expression evaluation problem.