

Data Structures HW2

Arrays, Linked Lists, Stacks and Queues

October 20, 2020

Write Pseudo-codes or explanations for the given problems below.

Problem 1 : Show You Can Reverse the order of the Nodes in a Linked List using only a single Unit of memory called '**aux**' (auxiliary).

(p.n. you know the Length of the linked list and you can use units of memory for '**while**' and '**for**' loops.)

Problem 2 : A Sparse Matrix is a matrix which more than half of its cells are 0, assume we have a Sparse Matrix with only the values of '0' and '1', show there can be other representations of the sparse matrix where we conserve memory usage.

(With the usage of Arrays and/or Linked Lists.)

$$\mathbf{A}_{sparse} = \begin{bmatrix} 0 & A_{12} & A_{13} & 0 & 0 \\ 0 & A_{22} & 0 & 0 & 0 \\ 0 & 0 & A_{33} & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & 0 \\ 0 & A_{52} & 0 & 0 & 0 \end{bmatrix}$$

Problem 3 : Show that you can Reverse the Order of the cells within an Array using a single Stack.

Problem 4 : Consider the following sequence of stack operations :

push(d) , push(h) , pop() , push(f) , push(s) , pop() , pop() , push(m).

(A) Assume the stack was initially empty, what is the sequence of popped elements (from last to first) and the sequence of elements within the stack.

(B) Assume we exchange the 'PUSH' and 'POP' operations with 'ENQUEUE' and 'DEQUEUE' and we have an empty Queue. what would be the sequence of Dequeued elements (from last to first) and the sequence of elements within the queue.

Problem 5 : Suppose you have a stack in which the values of 1 to 5 should be pushed into the stack in that order, but an item on top of the stack can be popped at any time.

give the sequence of 'PUSH' and 'POP' operations so that the order of the popped elements would be as described below :

- (a) 2, 4, 5, 3, 1.
- (b) 1, 3, 5, 4, 2.

BONUS PROBLEM :

Assume you have a queue with operations: `enqueue()`, `dequeue()`, `isEmpty()`.

How would you use the queue methods to simulate a stack, with `push()` and `pop()` operations?

(you can use multiple Queues)