CS2040C Semester 1 2018/2019

Data Structures and Algorithms

Tutorial 03 - Linked List, Stack, Queue, Deque

For Week 5 (Week Starting 10 September 2018)

Document is last modified on: September 5, 2018

1 Introduction and Objective

For this tutorial, you will need to review `https://visualgo.net/en/list?slide=1` (to last slide 9-5) about Linked List and all its variations as they will be the focus of today's tutorial.

2 Tutorial 03 Questions

Linked List, Mini Experiment

Q1). Please use the 'Exploration Mode' of `https://visualgo.net/en/list` to complete the following table (some cells are already filled as illustration). You can use use the mode selector at the top to change between (Singly) Linked List (LL), Stack, Queue, Doubly Linked List (DLL), and Deque mode. You can use 'Create' menu to create input array of various types. Stack, Queue and Deque are ADTs with a subset of list features. Hence, they can be implemented using a Linked List.

Mode →	Singly	Stack	Queue	Doubly	Deque
↓ Action	Linked List			Linked List	
search(any-v)	$O(N)$	not allowed	not allowed	$O(N)$	not allowed
peek-front()	$O(1)$				
peek-back()					$O(1)$
insert(0, new-v)				$O(1)$	
insert(N, new-v)					$O(1)$
insert(i, new-v), $i \in$[1..N-1]		not allowed			
remove(0)			not allowed		
remove(N-1)		not allowed			
remove(i), $i \in$[1..N-2]				$O(N)$	

You will need to fully understand the individual strengths and weaknesses of each Linked List variations discussed in class in order to be able to complete this mini experiment properly. You can assume that all Linked List implementations have head and tail pointers, have next pointers, and only for DLL and Deque: have prev pointers.

Q2). Assuming that we have a List ADT that is implemented using a Singly Linked List with both head and tail pointers. Show how to implement an additional operation `reverseList()` that takes in the current list of N items $\{a_0, a_1, ..., a_{N-2}, a_{N-1}\}$ and reverse it so that we have the reverse content $\{a_{N-1}, a_{N-2}, ..., a_1, a_0\}$. What is the time complexity of your implementation? Can you do this faster than $O(N)$?

Online Judge Exercises

Q3) How would you solve the Online Judge exercises mentioned in `https://visualgo.net/en/list?slide=9-5` using a Linked List (or other data structures). What methods of a Linked List would be used? What is the total time complexity of your algorithm?

- Kattis - Backspace (`https://open.kattis.com/problems/backspace`)
- UVA - Broken Keyboard (`https://uva.onlinejudge.org/external/119/11988.pdf`)
- Kattis - Integer Lists (`https://open.kattis.com/problems/integerlists`)

Problem Set 2

We will end the tutorial with discussion of PS2.