

IT5003 Mar-May 2024
Data Structures and Algorithms

Tutorial+Lab 03
Linked List, Stack, Queue, Deque

Document is last modified on: January 5, 2024

1 Introduction and Objective

For this tutorial, you will need to (re-)review <https://visualgo.net/en/list?slide=1> (to last slide 9-6) about List ADT and most of its variations (SLL, Stack, Queue, DLL, and Deque) as they will be the focus of today's tutorial.

2 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 the mode selector at the top to change between (Singly) Linked List (LL), Stack, Queue, Doubly Linked List (DLL), or Deque mode. You can use 'Create' menu to create input list of various types.

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

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 two additional operation:

1. `reverseList()` that takes in the current list of N items $\{a_0, a_1, \dots, a_{N-2}, a_{N-1}\}$ and reverse it so that we have the reverse content $\{a_{N-1}, a_{N-2}, \dots, a_1, a_0\}$. What is the time complexity of your implementation? Can you do this faster than $O(N)$?
2. `sortList()` that takes in the current list of N items and sort them so that $a_0 \leq a_1 \leq \dots \leq a_{N-2} \leq a_{N-1}$. What is the time complexity of your implementation? Can you do this faster than $O(N \log N)$?

Python Implementations: list, stack, and queue (deque)

Q3). To strengthen the discussions in the lecture, answer the following sub-questions:

a). What is Python list really is? Is it a Singly Linked List? A Doubly Linked List? or?

See <https://docs.python.org/3/tutorial/datastructures.html#more-on-lists>

b). How to implement an efficient Stack in Python?

See <https://docs.python.org/3/tutorial/datastructures.html#using-lists-as-stacks>

c). How to implement an efficient Queue in Python?

See <https://docs.python.org/3/tutorial/datastructures.html#using-lists-as-queues>

Hands-on 3

TA will run the second half of this session with a few to do list:

- PS2 Quick Debrief,
- Do a sample speed run of VisuAlgo online quiz that are applicable so far, e.g., <https://visualgo.net/training?diff=Medium&n=5&t1=5&module=list>.
- Finally, live solve another chosen Kattis problem involving a List.

Problem Set 3

We will end the tutorial with high level discussion of PS3 A+B.