

CS 1.2: Intro to Data Structures & Algorithms

Hash Table Time Complexity Worksheet

Name: _____

Given: Linked List Solutions – implementation and time complexity

The variable n represents the number of items stored in the list (or equivalently, number of nodes).

<i>Linked List operation</i>	<i><u>short summary in pseudocode (English) of the major steps performed in the implementation</u></i>	<i><u>best case running time</u></i>	<i><u>worst case running time</u></i>
is_empty	check if head node exists (None or not None)	$O(1)$	$O(1)$
length	traverse all nodes; count 1 for each node	$O(n)$	$O(n)$
append	add new node to end (after tail node); update tail property to point to new node	$O(1)$	$O(1)$
prepend	add new node to beginning (before head node); update head property to point to new node	$O(1)$	$O(1)$
find	traverse all nodes until matching data is found; if found, return matching data; if not, return None	$O(1)$	$O(n)$
delete	traverse all nodes until matching data is found; if found, set previous node to point to next node	$O(1)$	$O(n)$

New: Hash Table Operations – implementation and time complexity

Use the variable n for the number of key-value entries stored and b for the number of buckets.

<i>Hash Table operation</i>	<i><u>short summary in pseudocode (English) of the major steps performed in the implementation</u></i>	<i><u>best case running time</u></i>	<i><u>average case running time</u></i>
length			
items			
contains			
get			
set			
delete			