

CS 1.2: Intro to Data Structures & Algorithms

Hash Table Time Complexity Worksheet

Name: Mark Frazier

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>short summary in pseudocode (English) of the major steps performed in the implementation</i>	<i>best case running time</i>	<i>worst case running time</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>short summary in pseudocode (English) of the major steps performed in the implementation</i>	<i>best case running time</i>	<i>average case running time</i>
length	Iterate through all the buckets to count the number of key-value pairs	$O(n)$	$O(n)$
items	Iterate through all the buckets and then return a list of all the key-value pairs	$O(n)$	$O(n)$
contains	Access the correct bucket via the hash and then iterate until key is found. Return true if it contains the key, False if not. ($O(1)$ if minimal collisions)	$O(1)$	$O(1)$
get	Access the correct bucket via the hash and then iterate through the linked list and return value associated with the given key.	$O(1)$	$O(1)$
set	Access the correct bucket via the hash and then iterate through the linked list until given key is found. If found, delete the item, then append the new item.	$O(1)$	$O(1)$
delete	Access the correct bucket via the hash and then iterate through the linked list until given key is found. If found, delete the item.	$O(1)$	$O(1)$