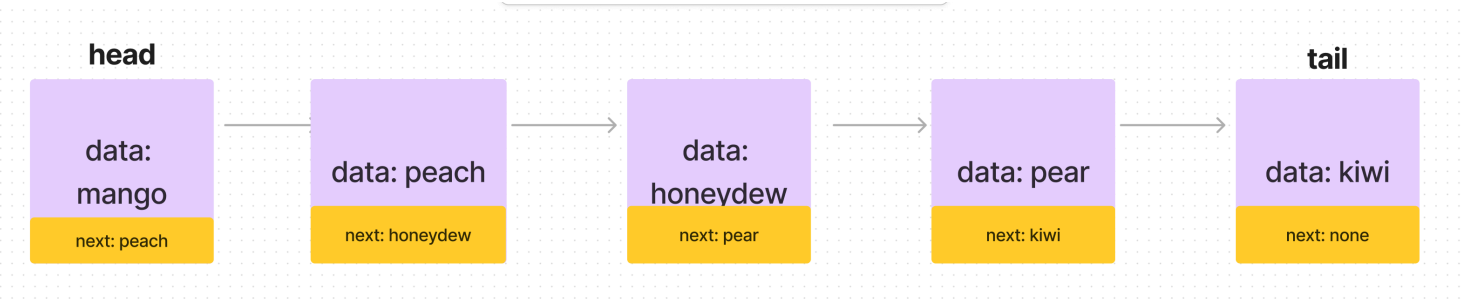
CS 1.2: Intro to Data Structures & Algorithms

Linked List Time Complexity Worksheet

Name: Marco Yip

Linked List Diagram – organization of data structure in memory

Draw a diagram of how a linked list data structure is organized in memory using references. The linked list should contain exactly 5 items: `'mango'`, `'peach'`, `'honeydew'`, `'pear'`, and `'kiwi'`.
Label the `head`, `tail`, `data`, and `next` properties in appropriate places to complete the diagram.



Linked List Operations – implementation and time complexity

Using your diagram above to guide you, complete the table below. First, write a short summary in pseudocode (English) of the major steps performed in the implementation of each operation. Then, analyze each operation's best case and worst case time complexity using big-O notation. Use the variable ***n*** for the number of items stored in the list (equivalently, the number of nodes).

Linked List operation	*short summary in pseudocode (English) of the major steps performed in the implementation*	*best case running time*	*worst case running time*
`is_empty`	Will check if a head node exists.	O(1)	O(1)
`length`	Will iterate through all nodes, adds 1 to count, and returns the count.	O(n)	O(n)
`append`	Adds the new node after the tail node and updates new node as the new tail.	O(1)	O(1)
`prepend`	Adds new node before the head node and updates new node as new head.	O(1)	O(1)
`find`	Iterates through the linked list and returns matching data if found. If not, returns none.	O(1)	O(n)
`delete`	Iterates through the linked list until the matching node is found. If found, set next node as the previous node's next node.	O(1)	O(n)