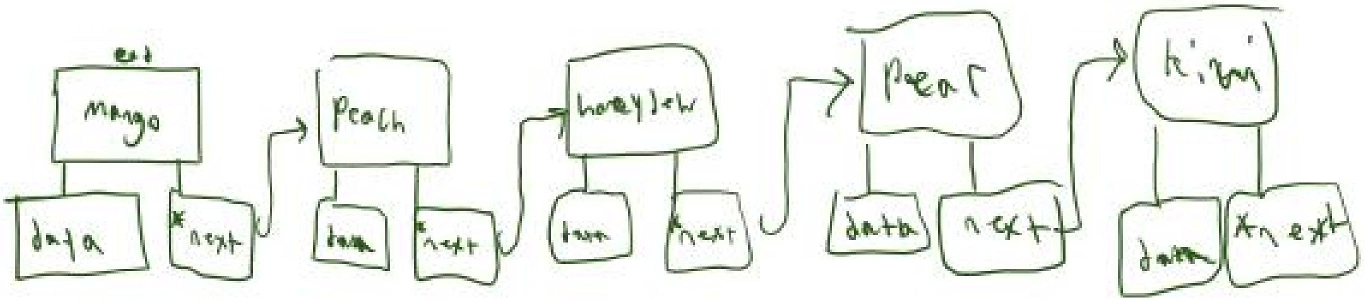
CS 1.2: Intro to Data Structures & Algorithms

Linked List Time Complexity Worksheet

Name: Chris

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`	Check if the head is empty	O(1)	O(1)
`length`	Traverse the list and add one to a count var until you reach the tail	O(n)	O(n)
`append`	Create a new node set it to the tail set the old tail's next to the new tail	O(1)	O(1)
`prepend`	Create a new node set it to the head and next is the old head	O(1)	O(1)
`find`	Traverse the list and check if the current node is what you are looking for then return that node	O(1)	O(n)
`delete`	Traverse the list and check if the current node is what you are looking for then set the prev node next to the current node's next	O(1)	O(n)