Sorting Customer Orders

**Bubble Sort** has a time complexity of O(n^2), which makes it inefficient for large datasets.

**Quick Sort** has an average time complexity of O(n log n), making it much more efficient for large datasets.

Quick Sort is generally preferred over Bubble Sort because of its superior time complexity and efficiency. While Bubble Sort is easier to implement and understand, its quadratic time complexity makes it impractical for large datasets. Quick Sort, on the other hand, uses a divide-and-conquer strategy, which significantly reduces the time complexity and makes it suitable for handling large datasets.

Employee Management System

Memory representation of an array is contiguous, meaning that all elements are stored sequentially in memory.

**Time Complexity**

* **Add**: O(1) on average, but O(n) in the worst case if resizing the array is required.
* **Search**: O(n) since we might need to check each element.
* **Traverse**: O(n) as we visit each element once.
* **Delete**: O(n) since we might need to find the element first, then possibly shift elements.

**Limitations of Arrays**

* **Fixed Size**: Once created, the size of an array cannot be changed. Dynamic resizing is costly (O(n) time complexity).
* **Inefficient for Search/Delete**: Searching and deleting elements can be inefficient (O(n) time complexity).
* **Better for Fixed-size Collections**: Arrays are more suitable for collections of fixed size where elements are frequently accessed by index.

Task Management System

**Singly Linked List**

A Singly Linked List is a collection of nodes where each node contains a data part and a reference (or link) to the next node in the sequence. The last node in the list points to null, indicating the end of the list.

**Doubly Linked List**

A Doubly Linked List is similar to a Singly Linked List but with an additional link. Each node contains a reference to the next node and a reference to the previous node, allowing traversal in both directions.

**Time Complexity**

* **Add**: O(n) in the worst case, as it requires traversing the list to the end to add the new node.
* **Search**: O(n), as it may require checking each node.
* **Traverse**: O(n), since it involves visiting each node.
* **Delete**: O(n), as it may require searching for the node to be deleted.

**Advantages of Linked Lists Over Arrays**

* **Dynamic Size**: Linked lists can grow and shrink dynamically, while arrays have a fixed size.
* **Efficient Insertions/Deletions**: Linked lists allow efficient insertions and deletions at any position, as they do not require shifting elements as in arrays.

Library Management System

**Linear Search**

* Linear search is the simplest search algorithm. It traverses the array or list sequentially to find the target value.
* Time Complexity: O(n) where n is the number of elements in the list.

**Binary Search**

* Binary search is more efficient but requires the array or list to be sorted. It divides the search interval in half repeatedly to find the target value.
* Time Complexity: O(log n) where n is the number of elements in the list.

Financial Forecasting

Recursion is a technique where a function calls itself to solve smaller instances of the same problem. Recursive algorithms can simplify problems by breaking them down into smaller, more manageable subproblems.

* Base case: The condition under which the recursion stops.
* Recursive case: The part of the function where the function calls itself.

The formula to calculate future value is:

