A queue is a fundamental data structure that follows the FIFO (First-In-First-Out) principle. It allows access to the element at the front of the queue. Common operations on queues include enqueue (add an item to the back), dequeue (remove the item from the front), and peek (retrieve the item at the front without removing it).

Queues can be efficiently implemented using singly linked lists. Singly linked lists provide O(1) time complexity for enqueue and dequeue operations because we only remove items from the front.

There are other types of queues such as priority queues and double-ended queues (deque). Priority queues use a custom comparer to determine the order of elements based on priority. Heaps are a good data structure to implement priority queues. Deques allow access to elements from both the front and the back. They can be implemented using doubly linked lists.