In computer science, a queue is a collection of data that is maintained in a sequence and can be modified by the addition of data at front of the sequence and the removal of data from the back of the sequence. Conventionally, the end of the sequence where data is added is called the tail or rear of the queue and the end at which where data is removed is called the head or front of the queue. Adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. A peek operation is often included that returns the next element in the queue without dequeuing it. A queue is a first-in-first-out (FIFO) data structure. In a FIFO data structure, the first element added to the queue will be the first one to be removed.

Queues provide services in computer science, transport, and operations research where various entities such as data, objects, persons, or events are stored and held to be processed later. In these contexts, the queue performs the function of a buffer. Another usage of queues is in the implementation of breadth-first search. (Queue (abstract data type), 2023)

Chart, waterfall chart

Description automatically generated

A hash function is any function that can be used to map data of arbitrary size to fixed-size values. The values returned by a hash function are called hash values. The values are usually used to index a fixed-size table called a hash table. Use of a hash function to index a hash table is called hashing.

Hash functions and their associated hash tables are used in data storage and retrieval applications to access data in a small and nearly constant time per retrieval. The amount of storage space required is only fractionally greater than the total space required for the data or records themselves. (Hash function, 2023)



In computer science, a binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child. In computing, binary trees are used in two different ways. First, as a means of accessing nodes based on the data associated with each node. Binary trees implemented this way are used to implement binary search trees and binary heaps and are used for efficient searching and sorting. Second, as a representation of data with a branching structure. The arrangement of nodes under and/or to the left or right of other nodes is a part of the information, therefore changing it would change the meaning. A common example is Huffman coding. (Binary tree, 2023)

There are many different types of binary trees. A binary tree is full if every node has either zero or two children. A binary tree is Complete if all the levels are filled except for the last level. The last level has all the nodes as far left as possible. A binary tree is Degenerate if all the nodes have a single child. A Degenerate binary tree has the same performance as a linked list. A binary tree is Perfect if every internal node has exactly two children and all the leaf nodes are at the same level. A binary tree is Balanced if the subtrees of every node differ in height by no more than one. (Types of Binary Tree, 2023)

In conclusion, queues, hashing and binary trees are important to the computer science field.

Graphical user interface

Description automatically generated

# References

*Binary tree*. (2023, March 1). Retrieved April 10, 2023, from Wikipedia: https://en.wikipedia.org/wiki/Binary\_tree

*Hash function*. (2023, March 3). Retrieved April 10, 2023, from Wikipedia: https://en.wikipedia.org/wiki/Hash\_function

*Queue (abstract data type)*. (2023, February 18). Retrieved April 10, 2023, from Wikipedia: https://en.wikipedia.org/wiki/Queue\_(abstract\_data\_type)