

Abstract Data Structures

Author: Rohith Perumalla

Date: 10/13/16

Subject: Computer Engineering

Citations:

"Collection (Java Platform SE 8)." Collection (Java Platform SE 8). Oracle, n.d. Web. 13 Oct. 2016.

"Set (Java Platform SE 8)." Set (Java Platform SE 8). Oracle, n.d. Web. 13 Oct. 2016.

Tutorialspoint.com. "Data Structure and Algorithms (DSA) Tutorial." www.tutorialspoint.com.

Tutorials Point, n.d. Web. 13 Oct. 2016.

Summary:

Data structure are systematic ways of organizing data in order to use it efficiently, data structures have an interface and an implementation. Data structures are often characterized by their correctness, time complexity, and space complexity. There are a few abstract data structures including: Collections, Lists, Stack, Queue, Sets, and more. Other data structures include: Linked Lists, Doubly Linked Lists, Circular Linked Lists, XOR Linked Lists, Trees, Heaps, and more. Even though there are many data structures all of them were designed to make storing data more efficient.

Analysis:

A data structure is used to efficiently store and access data. Data structures either work as a Contiguous-based structure or a Node-based structure. A Contiguous-based structure works as a lists of things while a Node-based structure works as a hierarchy. In computer science with object oriented programming (OOP) there are abstract classes are objects that can function as a base for similar classes and be inherited but, cannot be instantiated. Due to the inability to optimize all operations simultaneously, specialized data structures are designed for different situations. Data structures are specialized formats to efficiently store and access data, and depending on its specialization their best uses can vary.

Abstract Data Structures

One abstract data structure is a Collection (a.k.a. Container), a Collection is a group of n data elements that are grouped by some shared significance. Collections are abstract therefore cannot be instantiated but their flexibility allows them to be specialized. Collections can be modified to allow or disallow duplicate elements and can even be specialized to be ordered or unordered. Due to their flexibility Collections are inherited by many data structures including: Lists, Stacks, Queues, Sets, Trees, and other abstract and non-abstract data structures.

A List is an abstract contiguous data structure that inherits Collection in Java. Lists represents n elements, in Lists elements can occur more than once. This provides a way to store data as a countable sequence of values, which allows for easy access of elements. Lists are random access data structures allowing a user to add and remove elements from the list regardless of the position in the List, increasing the flexibility; however, being random access makes them more susceptible to corruption. Lists like collections are also versatile due to their abstract nature allowing them to be specialized even further. Some data structures that inherit Lists are Linked Lists, Doubly Linked Lists, Circular Linked Lists, XOR Linked Lists, and more.

Stacks and Queues are abstract contiguous data structures that inherit Collection in Java. Stacks and Queues both represent n elements in a sequential ordered collection. However unlike Lists Stacks and Queues are not random access they are open only on the ends. Stacks are Last In First Out (LIFO) structures meaning that the last element placed in the Stack is the first one seen if popped or accessed. LIFO structures can be represented by a stack plates the last plate placed in the stack is the first one removed for use. Due to Stacks LIFO nature they could be used in language processing to create space for parameters and local variables. Queues are persistent and First In First Out (FIFO) structures meaning the first element placed in the queue is the first one popped or access. FIFO structures can be represented by a line at a grocery store where the customer in line first is the first one checked out. Due to Queues FIFO nature they have the potential to be used in e-commerce where customers who purchase goods first will get their credentials processed first. Stacks and Queues like Collections and Lists are also flexible due to their abstract nature allowing them to be specialized to accomplish more custom tasks. Overall Stacks are often used for syntax parsing, backtracking, runtime memory management and Queues can be used for functional implementation due to its persistent nature.

Abstract Data Structures

Sets are abstract contiguous data structures that inherit Collection in Java. A Set is a sequential collection of elements. Sets ensure that when a value is placed in a Set that there are no duplicates, so if a value already exists in the Set nothing is done, but if it does not contain it, it is added to the Set. This could be used in constructive digital graphic design with a workspace having a specific number of pixels, each pixel with a distinct location. Sets are also versatile due to their abstract nature allowing them to be specialized even further. Some data structures that inherit Sets are Navigable Sets, Sorted Sets, Enum Sets, Hash Sets, Tree Sets, and more.

Data structures are systematic ways of organizing data in order to use it efficiently, and abstract structures allow customization with type of access: random access, LIFO, or FIFO; or with type of storage: with or without duplicates. Programmers can specialize abstract data structures and the specialization allows the highest efficiency depending on the problem they being solved. Uses ranging from storing customer credentials on E-Commerce to storing workspace data for graphic design software. Altogether, abstract data structures are extremely useful due to their overall versatility.