Data Structures Term#Definition

Abstract data type (ADT)#A data type whose properties (domain and operations) are specified independently of any particular implementation

Abstract method#A method declared in a class or an interface without a method body

Abstraction#A model of a system that includes only the details essential to the perspective of the viewer of the system

Access modifier#Indicates the availability of a Java construct: public, protected, package or private

Activation record (stack frame)#Space used at run time to store information about a method call, including the parameters, local variables, and return address

Adjacency list#A linked list that identifies all the vertices to which a particular vertex is connected; each vertex has its own adjacency list

Adjacency matrix#For a graph with N nodes, an N by N table that shows the existence (and weights) of all edges in the graph

Adjacent from# If vertex A is connected to vertex B in a directed graph by an edge (A, B) then B is adjacent from A

Adjacent to# If vertex A is connected to vertex B in a directed graph by an edge (A, B) then A is adjacent to B

Adjacent vertices#Two vertices in a graph that are connected by an edge

Algorithm#A sequence of unambiguous instructions that solve a problem, within a finite amount of time, given a set of valid input

Alias When two variables reference the same object, they are aliases of each other

Amortized analysis#Using average case analysis in place of worst case analysis to spread the cost of unusual extra processing that is only incurred occasionally

Ancestor#A parent of a node, or a parent of an ancestor in a tree

Anonymous inner class#An anonymous class is a class without a name; instead of defining a class in one place and then instantiating it somewhere else, as is the usual approach, we can define a class exactly at the place in the code where it is being instantiated; because it is created at the same place it is defined, there is no need for a class name

Application (Java application)#The part of a Java program where processing begins; the class of a Java program that contains the main method

Average case complexity#Related to the average number of steps required by an algorithm, calculated across all possible sets of input values

Bag#A Collection-like ADT that also provides operations grab, count, removeAll, and clear

Base case#The case for which the solution can be stated nonrecursively

Best case complexity#Related to the minimum number of steps required by an algorithm, given an ideal set of input values, in terms of efficiency

Binary search#A search approach that eliminates half the remaining possibilities at each step

Binary search tree#A binary tree in which the value in any node is greater than or equal to the value in its left child and any of its descendants (the nodes in the left subtree) and less than the value in its right child and any of its descendants (the nodes in the right subtree)

Binary tree#A tree in which each node is capable of having two child nodes: a left child node and a right child node

Breadth-first traversal (level order traversal)#A tree traversal that first visits the root of the tree, then next visits, in turn, the children of the root (typically from leftmost to rightmost), followed by visiting the children of the children of the root and so on until all of the nodes have been visited

Bucket#A collection of elements associated with a particular hash location

Bushy#A well balanced tree

Chain#A linked list of elements that share the same hash location

Checked exception#Java exception that when thrown, must be either caught by the surrounding code or rethrown by the surrounding method

Children#The successor nodes of a node in a tree

Circular linked list#A list in which every node has a successor; the “last” element is succeeded by the “first” element

Client#A client of a class/ADT is any code that uses the class/ADT, be it an application or another ADT/class

Clustering#The tendency of elements to become unevenly distributed in the hash table, with many adjacent locations containing elements

Collection#An object that holds other objects; typically we are interested in inserting, removing, searching, and iterating through the contents of a collection

Collections Framework (Java Collections Frameworks)#The set of classes and interfaces that implement Collection ADTs

Collision#The condition resulting when two or more keys produce the same hash location

Complete binary tree#A binary tree that is either full or full through the next-to-last level, with the leaves on the last level as far to the left as possible

Complete graph#A graph in which every vertex is directly connected to every other vertex

Compression function#A function that "compresses" the wider domain of numbers, representing the hash codes of the elements being inserted into a hash table, into the smaller range of numbers, representing the indices of the hash table

Concurrent programs#Several interacting code sequences are executing simultaneously, possibly through an interleaving of their statements by a single processor, possibly through execution on distinct processors

Connected component#A maximal set of vertices of a graph that are connected to each other, along with the edges associated with those vertices

Connected graph#A graph that consists of a single connected component

Connected vertices#In an undirected graph we say that two vertices are connected if there is a path between them

Constructor#An operation that creates a new instance of a class

Cycle#A path in a graph that begins and ends at the same vertex

Data abstraction#The separation of a data type’s logical properties from its implementation

Data encapsulation#A programming language feature that enforces information hiding

Deallocate#To return the storage space for an object to the pool of free memory so that it can be reallocated to new objects

Depth-first traversal#A tree traversal that first visits the root of the tree, then next traverses as far as possible along the leftmost path, until reaching a leaf, and then "backing up" as little as needed before traversing again down to a leaf and so on until all of the nodes have been visited

Depth of the recursion#The number of activation records on the system stack associated with a given recursive method

Deque (double-ended queue)#A linear structure that only allows access (insertion/removal of elements) at its ends, i.e. at its front and at its rear

Descendant#A child of a node, or a child of a descendant in a tree

Direct recursion#Recursion in which a method directly calls itself

Directed graph (digraph)#A graph in which each edge is directed from one vertex to another (or the same) vertex

Disconnected graph#A graph that is not connected

Double-ended queue (deque)#A linear structure that only allows access (insertion/removal of elements) at its ends, that is, at its front and at its rear

Doubly linked list#A linked list in which each node is linked to both its successor and its predecessor

Dynamic (run-time) binding#When the association between a variable or method and the actual memory location containing the variable or method is made during the execution of a program

Dynamic memory management#The allocation and deallocation of storage space as needed while an application is executing

Edge (arc)#A pair of vertices representing a connection between the two vertices in a graph

Exception#Associated with an unusual, sometimes unpredictable event, detectable by software or hardware, which requires special processing; the event may or may not be erroneous

Factory method#A method that creates and returns objects

Full binary tree#A binary tree in which all of the leaves are on the same level and every nonleaf node has two children

Garbage#The set of currently unreachable objects

Garbage collection#The process of finding all unreachable objects and deallocating their storage space

General (recursive) case#The case for which the solution is expressed in terms of a smaller version of itself

Generics#Parameterized types; allow us to define a set of operations that manipulate objects of a particular class, without specifying the class of the objects being manipulated until a later time

Graph#A data structure that consists of a set of vertices and a set of edges that relate the vertices to each other

Hash code#The output of the hash function that is associated with the input object

Hash function#A function that takes as input the key of an element and produces an integer as output

Hash table#The data structure used to store elements using hashing

Hashing#The technique used for ordering and accessing elements in a collection in a relatively constant amount of time by manipulating the element's key to identify the element's location in the collection

Header node#A placeholder node at the beginning of a list; used to simplify list processing

Heap#An implementation of a priority queue based on a complete binary tree, each of whose elements contains a value that is greater than or equal to the value of each of its children

Height#The maximum level of a tree

Hybrid data structure#A synergetic combination of two data structures

Immutable object#An object that once instantiated cannot be changed

Indirect recursion#See Recursion (indirect)

Information hiding#The practice of hiding details within a module with the goal of controlling access to the details from the rest of the system

Inheritance of classes#A Java class can extend one other Java class, inheriting its attributes and methods

Inheritance of interfaces#A Java interface can extend another Java interface, inheriting its requirements; if interface B extends interface A, then classes that implement interface B must also implement interface A; usually, interface B adds abstract methods to those required by interface A

Inheritance tree#A tree, rooted in the Object class, representing all of the inherited class relationships between Java classes

Inorder traversal#A systematic way of visiting all the nodes in a binary tree by visiting the nodes in the left subtree of a node, then visiting the node, and then visiting the nodes in the right subtree of the node

Instantiation#Using the new command to create a new instance/object of a Java class

Interior node#A tree node that is not a leaf

Key#The attributes that are used to determine the identity and logical order of the elements in a collection

Leaf#A tree node that has no children

Level#The level of a tree node is its distance from the root (the number of connections between itself and the root)

Level order traversal (breadth-first traversal)#A tree traversal that first visits the root of the tree, then next visits, in turn, the children of the root (typically from leftmost to rightmost), followed by visiting the children of the children of the root and so on until all of the nodes have been visited

Linear probing#Resolving a hash collision by sequentially searching a hash table beginning at the location returned by the hash/compression function

Linear relationship#Each element except the first has a unique predecessor, and each element except the last has a unique successor

List#A collection that exhibits a linear relationship among its elements

Load threshold#When the ratio of entries to total space in a hash table rises above the load threshold the size of the hash table is increased

Map#An ADT that associates keys with unique values

Methodology#A collection of specific procedures for creating a software system to meet a user’s needs

Multiple inheritance of interfaces#A Java interface may extend more than one interface; if interface C extends both interface A and interface B, then classes that implement interface C must also implement both interface A and interface B; sometimes multiple inheritance of interfaces are used simply to combine the requirements of two interfaces, without adding any more methods

Multitasking#Perform more than one task at a time

Natural order#The order established by a class’s compareTo method

Observer#An operation that allows us to observe the state of an object without changing it

Optional operation/method#Some operations fit well with one implementation of an ADT but do not make sense for another implementation; in these cases we indicate in our interface that an operation is optional; implementations can elect not to support such operations

Order of growth complexity#A notation that expresses computing time (complexity) as the term in a function that increases most rapidly relative to the size of a problem

Order property of heaps#For every node in the underlying tree, the value stored in that node is greater than or equal to the value in each of its children

Override#Redefining a method of a superclass in a subclass

Parent#A tree node’s unique predecessor is its parent

Path#A sequence of vertices that connects two vertices in a graph

Polymorphism#The ability of an object variable to reference objects of different classes at different times during the execution of a program

Postconditions (effects)#The results expected at the exit of a method, assuming that the preconditions are true

Postorder traversal#A systematic way of visiting all the nodes in a binary tree by visiting the nodes in the left subtree of a node, then visiting the nodes in the right subtree of the node, and then visiting the node

Preconditions#Assumptions that must be true on entry into a method for it to work correctly

Preorder traversal#A systematic way of visiting all the nodes in a binary tree by visiting a node, then visiting the nodes in the left subtree of the node, and then visiting the nodes in the right subtree of the node

Primitive variable#A Java variable of one of the eight directly supported types (byte, char, short, int, long, float, double, and boolean) that stores its contents by-value, that is, the actual value of the variable is held in the memory location associated with the variable

Priority queue#An abstract data type where only the highest-priority element can be accessed/removed

Quadratic probing#Resolving a hash collision by using the rehashing formula (hash value +/–I2) % array-size

Queue#A structure in which elements are added to the rear and removed from the front; a “first in, first out” (FIFO) structure

Recursion (indirect)#See Indirect recursion

Recursive algorithm#A solution that is expressed in terms of (1) smaller instances of itself, and (2) a base case

Recursive call#A method call in which the method being called is the same as the one making the call

Recursive (general) case#The case for which the solution is expressed in terms of a smaller version of itself

Recursive definition#A definition in which something is defined in terms of smaller versions of itself

Reference variable#A Java variable associated with an object defined by a Java class—it stores its contents by-reference, that is, the variable holds the address where the object resides in memory

Rehash#Recomputing the locations for all of the elements in a hash table, for example, when the underlying hash table is resized

Root#The top node of a tree structure; a node with no parent

Run-time (dynamic) binding#When the association between a variable or method and the actual memory location associated with the variable or method is made during the execution of a program

Run-time (system) stack#A system data structure that keeps track of activation records during the execution of a program

Self-organizing (adjusting/balancing) tree#A tree that adjusts its pattern of nodes to keep itself “balanced” after nodes are added or removed

Self-referential class#A class that includes an instance variable or variables that can hold a reference to an object of the same class

Sequential search#A search approach that examines each element in a collection one by one sequentially

Set#A Collection class that does not allow duplicate elements

Shape property of heaps#The underlying tree must be a complete binary tree

Siblings#Tree nodes with the same parent

Signature#The distinguishing features of a method heading; the combination of a method name with the number and type(s) of its parameters in their given order

Skewed tree#A tree that is long and narrow; the opposite of a bushy/balanced tree

Snapshot#A copy of a data structure at some point in time

Stable sort#A sorting algorithm that preserves the order of duplicates

Stack#A structure in which elements are added and removed from only one end; a “last in, first out” (LIFO) structure

Subclass#If class A inherits from class B, then we say that "A is a subclass of B"

Subtree#A node and all of its descendants form a subtree rooted at the node

Superclass#If class A inherits from class B, then we say that "B is a superclass of A"

System (run-time) stack#A system data structure that keeps track of activation records during the execution of a program

Tail recursion#The case in which a method contains only a single recursive invocation and it is the last statement to be executed in the method

Test driver#A program that calls operations exported from a class, allowing us to test the results of the operations

Test harness#A stand-alone program designed to facilitate testing of the implementations of algorithms

Trailer node#A placeholder node at the end of a list; used to simplify list processing

Transformer#An operation that changes the internal state of an object

Tree#A structure with a unique starting node (the root), in which each node is capable of having multiple child nodes, and in which a unique path exists from the root to every other node

Unchecked exception#An exception of the RunTimeException class; it does not have to be explicitly handled by the method within which it might be raised

Undirected graph#A graph in which the edges have no direction

Unified Modeling Language (UML)#A collection of diagramming techniques used to describe software

Vertex#A node in a graph

Weighted graph#A graph in which each edge carries a value

Worst case complexity#Related to the maximum number of steps required by an algorithm, given the worst possible set of input values in terms of efficiency