Quizlet

Data Types

Study online at quizlet.com/_3e9f4
k

1. **8 Primitive Data Types** Boolean , byte , char , short , int , long , float and double.

2. **Abstract Data Type** In computer science, an abstract data type (ADT) is a mathematical model for data types where a data type is defined by its behavior (semantics) from the point of view of a user of the data, specifically in terms of possible values, possible operations on data of this type, and the behavior of these operations;

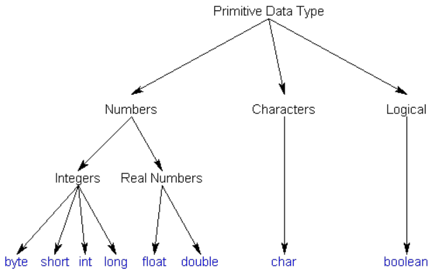
   An ADT involves both data and operations on that data

3. **Array** Array data structure, or simply an array, is a data structure consisting of a collection of elements (values or variables), each identified by at least one array index or key. An array is stored so that the position of each element can be computed from its index tuple by a mathematical formula.

4. **Boolean** Primitive: True or False

5. **Character** Primitive: Any letter, number, space, punctuation mark, or symbol that can be typed on a computer. The list of characters that can be typed is defined by the ASCII and extended ASCII set. ("a" or "%")

6. **Chart Breakdown of Primitive Data Types**



7. **Composite types or Non-primitive type** A variable of a non-primitive type doesn't contain the value directly; instead, it is a reference (similar to a pointer) to an object.

8. **Data Structure Review** **Types: Collection Container**

   Abstract: Associative array, Multimap, List, Stack, Queue, Double-ended queue, Priority queue, Double-ended priority queue, Set, Multiset, Disjoint-set

   Arrays: Bit array, Circular buffer, Dynamic array, Hash table, Hashed array tree, Sparse matrix

   Linked:Association list, Linked list, Skip list, Unrolled linked list, XOR linked list

   Trees: B-tree, Binary search tree, AA tree, AVL tree, Red-black tree, Self-balancing tree, Splay tree, Heap, Binary heap, Binomial heap, Fibonacci heap, R-tree, R* tree, R+ tree, Hilbert R-tree, Trie Hash tree

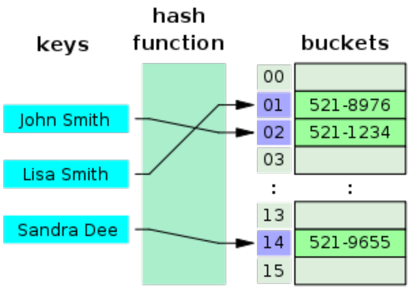
   Graphs: Binary decision diagram, Directed acyclic graph, Directed acyclic word graph

9. **Data Type** A particular kind of data item, as defined by the values it can take, the programming language used, or the operations that can be performed on it.

10. **Data type vs. Data structure** A **data structure** is an abstract description of a way of organizing data to allow certain operations on it to be performed efficiently. A **data type** is just a class of values that can be concretely constructed and represented.

11. **Deletion operation** Deletion means removal of a data element from a data structure if it is found.

12. **Dictionary or Hash Map**



In computer science, an associative array, map, symbol table, or dictionary is an abstract data type composed of a collection of (key, value) pairs, such that each possible key appears at most once in the collection.

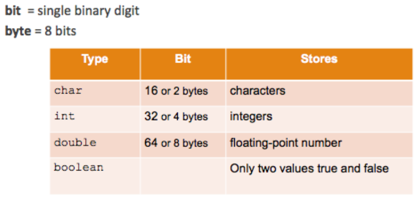
13. **Different representations of the array data type**

Array
Bit array
Bit field
Bitboard
Bitmap
Circular buffer
Control table
Image
Dope vector
Dynamic array
Gap buffer
Hashed array tree
Heightmap
Lookup table
Matrix
Parallel array
Sorted array
Sparse array
Sparse matrix
Iliffe vector
Variable-length array

14. **Different representations of the list data type**

Doubly linked list
Array list
Linked list
Self-organizing list
Skip list
Unrolled linked list
VList
Conc-Tree list
Xor linked list
Zipper
Doubly connected edge list
Difference list
Free list

15. **Double**



bit = single binary digit
byte = 8 bits

Type	Bit	Stores
char	16 or 2 bytes	characters
int	32 or 4 bytes	integers
double	64 or 8 bytes	floating-point number
boolean		Only two values true and false

Primitive: A wider floating-point size (e.g. 1.79769313486232E308)

16. **Examples of non-primitive data structures**

Array, Tuple, Class

17. **Floating-point**

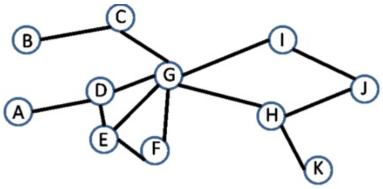
Primitive: Single-precision real number values;

The term floating point refers to the fact that a number's radix point (decimal point, or, more commonly in computers, binary point) can "float"; that is, it can be placed anywhere relative to the significant digits of the number.

Single-precision floating-point format is a computer number format that occupies 4 bytes (32 bits) in computer memory and represents a wide dynamic range of values by using a floating point. In IEEE 754-2008 the 32-bit base-2 format is officially referred to as binary32. It was called single in IEEE 754-1985.

Simply put, 12.34 is a float value, and 1234 is an integer

18. **Graph**



In computer science, a graph is an abstract data type that is meant to implement the undirected graph and directed graph concepts from mathematics.

A graph data structure consists of a finite (and possibly mutable) set of vertices or nodes or points, together with a set of unordered pairs of these vertices for an undirected graph or a set of ordered pairs for a directed graph. These pairs are known as edges, arcs, or lines for an undirected graph and as arrows, directed edges, directed arcs, or directed lines for a directed graph. The vertices may be part of the graph structure, or may be external entities represented by integer indices or references.

A graph data structure may also associate to each edge some edge value, such as a symbolic label or a numeric attribute (cost, capacity, length, etc.).

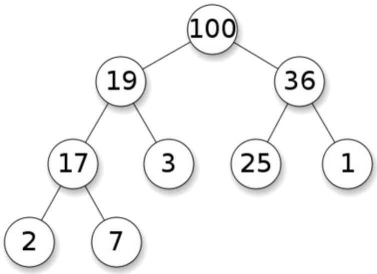
19. **Graph Types**

Graph
Adjacency list
Adjacency matrix
Graph-structured stack
Scene graph
Binary decision diagram
Zero-suppressed decision diagram
And-inverter graph
Directed graph
Directed acyclic graph
Propositional directed acyclic graph
Multigraph
Hypergraph

20. **Hash Types**

Bloom filter
Count-Min sketch
Distributed hash table
Double Hashing
Dynamic perfect hash table
Hash array mapped trie
Hash list
Hash table
Hash tree
Hash trie
Koorde
Prefix hash tree
Rolling hash
MinHash
Quotient filter
Ctrie

21. **Heap**



In computer science, a heap is a specialized tree-based data structure that satisfies the heap property: If A is a parent node of B, then the key (the value) of node A is ordered with respect to the key of node B with the same ordering applying across the heap.

22. **Insertion operation**

Insertion means addition of a new data element in a data structure.

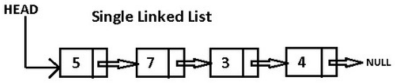
23. **Integer**

Primitive: Integral or fixed-precision values (e.g. 10, 42, 0, etc)

24. **Linear Data Structure**

A data structure is said to be linear if its elements form a sequence.

25. **Linked List**



In computer science, a linked list is a linear collection of data elements, called nodes, each pointing to the next node by means of a pointer. It is a data structure consisting of a group of nodes which together represent a sequence. ... Linked lists are among the simplest and most common data structures.

26. **List of ADTs**

Linear Data Structures:
Array,
Linked List,
Stack,
Queue,
Set,
Dictionary/Hash Map

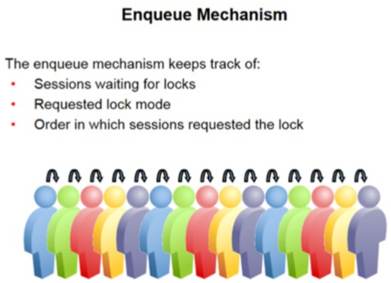
Nonlinear Data Structures:
Tree (binary tree, heap, etc),
Graph (digraph/undirected, cyclic/acyclic graph, etc)

27. **Merging operation**

Combining elements of two similar data structures to form a new data structure of the same type, is called merging.

28. **Primitive Type** The most basic data types available within a programming language; The building blocks of data manipulation; Such types serve only one purpose — containing pure, simple values of a kind

29. **Queue**



Enqueue Mechanism

The enqueue mechanism keeps track of:

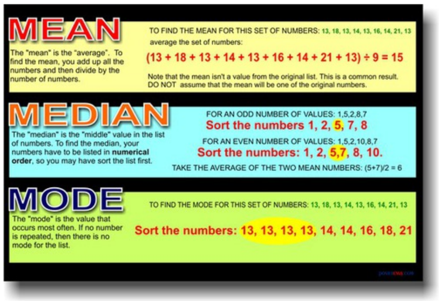
- Sessions waiting for locks
- Requested lock mode
- Order in which sessions requested the lock

A particular kind of ADT or collection in which the entities in the collection are kept in order and the principal (or only) operations on the collection are the addition of entities to the rear terminal position, known as enqueue, and removal of entities from the front terminal position, known as dequeue. This makes the queue 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. Often a peek or front operation is also entered, returning the value of the front element without dequeuing it. A queue is an example of a linear data structure, or more abstractly a sequential collection.

30. **Reference (also called a pointer or handle)** Primitive: A small value referring to another object's address in memory, possibly a much larger one.

31. **Searching operation** Searching involves searching for the specified data element in a data structure.

32. **Set**



MEAN

The "mean" is the "average". To find the mean, you add up all the numbers and then divide by the number of numbers.

TO FIND THE MEAN FOR THIS SET OF NUMBERS: 13, 18, 13, 14, 13, 16, 14, 21, 13

average the set of numbers:

(13 + 18 + 13 + 14 + 13 + 16 + 14 + 21 + 13) ÷ 9 = 15

Note that the mean isn't a value from the original list. This is a common result. DO NOT assume that the mean will be one of the original numbers.

MEDIAN

The "median" is the "middle" value in the list of numbers. To find the median, your numbers have to be listed in **numerical order**, so you may have sort the list first.

FOR AN ODD NUMBER OF VALUES: 1,5,2,8,7

Sort the numbers 1, 2, 5, 7, 8

FOR AN EVEN NUMBER OF VALUES: 1,5,2,10,8,7

Sort the numbers: 1, 2, 5,7, 8, 10.

TAKE THE AVERAGE OF THE TWO MEAN NUMBERS: (5+7)/2 = 6

MODE

The "mode" is the value that occurs most often. If no number is repeated, then there is no mode for the list.

TO FIND THE MODE FOR THIS SET OF NUMBERS: 13, 18, 13, 14, 13, 16, 14, 21, 13

Sort the numbers: 13, 13, 13, 13, 14, 14, 16, 18, 21

In computer science, a set is an abstract data type that can store certain values, without any particular order, and no repeated values. It is a computer implementation of the mathematical concept of a finite set.

33. **Sorting operation** Arranging data elements of a data structure in a specified order is called sorting.

34. **Stack**

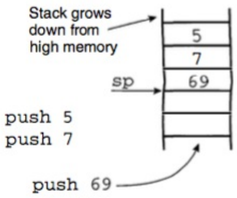


Figure 8.1 Stack data structure.

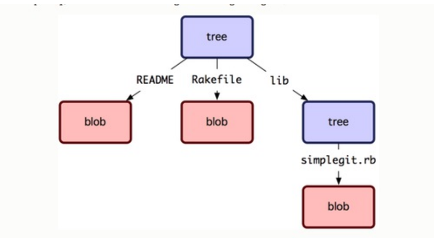
In computer science, a stack is an abstract data type that serves as a collection of elements, with two principal operations: push, which adds an element to the collection, and pop, which removes the most recently added element that was not yet removed.

35. **String** Primitive: A sequence of characters

36. **Subcategories of Tree ADT** **Binary trees**, B-Trees, **Heaps**, **Trees**, Multiway trees, Space-partitioning trees, Application-specific trees

37. **Traversal operation** Traversal of a data structure means processing all the data elements present in it.

38. **Tree**



In computer science, a tree is a widely used abstract data type (ADT)—or data structure implementing this ADT—that simulates a hierarchical tree structure, with a root value and subtrees of children with a parent node, represented as a set of linked nodes.

A tree data structure can be defined recursively (locally) as a collection of nodes (starting at a root node), where each node is a data structure consisting of a value, together with a list of references to nodes (the "children"), with the constraints that no reference is duplicated, and none points to the root.

Alternatively, a tree can be defined abstractly as a whole (globally) as an ordered tree, with a value assigned to each node. Both these perspectives are useful: while a tree can be analyzed mathematically as a whole, when actually represented as a data structure it is usually represented and worked with separately by node (rather than as a list of nodes and an adjacency list of edges between nodes, as one may represent a digraph, for instance). For example, looking at a tree as a whole, one can talk about "the parent node" of a given node, but in general as a data structure a given node only contains the list of its children, but does not contain a reference to its parent (if any).

39. **Void or Null** Primitive: No value (however, still **not** false)