Data structures

Python Reference · Vocabulary in context

Word	Example
abstract data type /ˈæbstrækt ˈdeɪtə taɪp/	A stack is an **abstract data type**: we know what it does, not how it is built.
stack /stæk/	The last item put on the **stack** is the first one taken off.
LIFO /ˈlaɪfəʊ/	A stack is **LIFO**: last in, first out.
push /pʊʃ/	To **push** means to add an item to the top of the stack.
pop /pɒp/	To **pop** means to take the top item off the stack.
queue /kjuː/	A **queue** is first in, first out.
FIFO /ˈfaɪfəʊ/	A queue is **FIFO**: first in, first out.
enqueue /enˈkjuː/	**Enqueue** adds an item to the back of the queue.
dequeue /diːˈkjuː/	**Dequeue** takes the item from the front of the queue.
linked list /lɪŋkt lɪst/	In a **linked list**, each item points to the next one.
node /nəʊd/	Each **node** holds a value and a pointer.
pointer /ˈpɔɪntə/	The **pointer** shows where the next item is.
hash table /hæʃ ˈteɪbl/	A Python dictionary is a **hash table**. Python
hash function /hæʃ ˈfʌŋkʃn/	A **hash function** turns a key into a position in the table.

exercises.lol exercises.com.cn

Word	Example
collision /kəˈlɪʒn/	A **collision** happens when two computers send at the same time.
binary search tree /ˈbaɪnəri sɜːtʃ triː/	In a **binary search tree**, smaller values go left and bigger values go right.
graph /græf/	A **graph** has nodes joined by edges.
vertex /ˈvɜːteks/	Each **vertex** in a graph can be joined to others by edges.
edge /edʒ/	Each **edge** in a graph joins two nodes.
adjacency list /əˈdʒeɪsənsi lɪst/	An **adjacency list** stores each node with a list of its neighbours.