List – a *sequence* (maintains positional information and allows positional access)

Duplicates OK

Operations (nominal)	ArrayList	LinkedList
boolean isEmpty()	O(1)	O(1)
int size()	O(1)	O(1)
boolean add(value)	amortized O(1)	O(1) [how?]
add (index, value)	O(n)	O(n)
boolean contains (value)	O(n)	O(n)
AnyType get(index)	O(1)	O(n)
AnyType set(index, value)	O(1)	O(n)
boolean remove(value)	O(n)	O(n)
Iterator operations		
hasNext()	O(1)	O(1)
next()	O(1)	O(1)
remove()	O(n)	O(1)

Set – unordered collection (no positional information; but there is TreeSet…)

No duplicates

Operations (nominal)	HashSet*	TreeSet
boolean isEmpty()	O(1)	O(1)
int size()	O(1)	O(1)
boolean add(value)	O(1)	O(log n)
boolean contains(value)	O(1)	O(log n)
boolean remove(value)	O(1)	O(log n)
Iterator operations		
hasNext()	O(1)	O(1)
next()	O(1)	initial call O(log n), [why?] thereafter O(1)
remove()	O(1)	O(log n)

Map – maps (unique) keys to values (1-1, 1-many if value is a collection)

Operations (nominal)	HashMap*	TreeMap
boolean isEmpty()	O(1)	O(1)
int size()	O(1)	O(1)
AnyType** put(key, value)	O(1)	O(log n)
boolean containsKey(key)	O(1)	O(log n)
AnyType get(key)	O(1)	O(log n)
AnyType remove(key)	O(1)	O(log n)
Set keySet()	O(k)	O(k)

* **BIG CAVEAT**: if you're using a HashMap or HashSet, the key's class MUST override hashCode() in a manner consistent with its override of equals()!

** seems odd for Map.put() to return a value… what value does Map.put() return?

Stack – access at one end (Last In – First Out)

Operations (nominal)	ArrayList	LinkedList
boolean isEmpty()	O(1)	O(1)
push(value)	O(1)*	O(1)*
AnyType pop()	O(1)*	O(1)*
AnyType peek()	O(1)	O(1)

* where does the front of the stack have to be to guarantee these performance results?

Queue – access at both ends (First In – First Out)

Operations (nominal)	ArrayList	LinkedList
boolean isEmpty()	O(1)	O(1)
enqueue(value)	O(1)	O(1)*
AnyType dequeue()	O(n)	O(1)*
AnyType peek()	O(1)	O(1)

* what has to be true of LinkedList to guarantee O(1) performance for both of these operations?

Priority Queue (assuming uniformly random distribution of inputs; ordered inputs will affect Big O)

Operations (nominal)	Min-Heap	Ordered List	Unordered List
boolean isEmpty()	O(1)	O(1)	O(1)
add(value)	O(log n)	O(n)	O(1)
AnyType peekMin()	O(1)	O(1)	O(n)
AnyType removeMin()	O(log n)	O(1)	O(n)