Winston Wang

Mr. Kuszmaul

Mapcar

Implement 'mapcar' in java on a collection of your choice. It is OK to fail in the general case --- you may settle for a specialization.

**HashSet**

Class/Interface/Abstract Class? Class

Search Time: Typically O(1), worst case is O(n)

Insertion Time: O(1)

Deletion Time: O(1)

Brief Description:

This set is very similar to TreeSet, except that HashSet is not ordered. No duplicates can exist

**Map**

Class/Interface/Abstract Class? Interface

Search Time: O(1) with high probability

Insertion Time: O(1) with high probability

Deletion Time: O(1) with high probabiliy

Brief Description:

This data structure contains objects that are linked to a key, much like locations are linked to coordinates on a real map.

**HashMap**

Class/Interface/Abstract Class? Class

Search Time: O(1) with high probability

Insertion Time: O(1) with high probability

Deletion Time: O(1) with high probabiliy

Brief Description:

This data structure contains objects that are linked to a key, much like locations are linked to coordinates on a real map. It is not that much different from Map, although HashMap can be instantiated.

**TreeMap**

Class/Interface/Abstract Class? Class

Search Time: Guaranteed O(log(n))

Insertion Time: Guaranteed O(log(n))

Deletion Time: Guaranteed O(log(n))

Brief Description:

This is a type of map, and the keys of the map sort it.

**Queue**

Class/Interface/Abstract Class? Interface

Search Time: O(n)

Insertion Time: O(1)

Deletion Time: O(1)

Brief Description:

Queues generally are first in first out, as opposed to stacks, which are first in last out.

**Stack**

Class/Interface/Abstract Class? Class

Search Time: O(n)

Insertion Time: Push? O(1)

Deletion Time: Pop? O(1 )

Brief Description:

A stack is a data structure that is first in last out. It has a couple different methods, including push and pop. The push method adds an element to the top of the stack, while pop takes the top element off.

**LinkedList**

Class/Interface/Abstract Class? Class

Search Time: O(n)

Insertion Time: O(n)

Deletion Time: O(n)

Brief Description:

A linked list is a data structure in which each node contains two fields, the data and the link to the next node.

**Collection**

Class/Interface/Abstract Class? Interface

Search Time: O(n)

Insertion Time: O(n)

Deletion Time: O(n)

Brief Description:

This is the root interface for the collection hierarchy, which means classes such as ArrayList, LinkedList, Queue, Stack, and Vector all implement Collection.