Basic Implementation of Dictionary/Map

We need some kind of collection to hold the keys/elements in the map.

There are two basic collections
- array
- linked list

and two basic ways elements can be ordered within those collections
- not sorted
- sorted

Basic Implementation of Map/Dictionary

Dictionary operation	Unsorted array	Sorted array	Singly linked		Doubly linked	
			unsorted	sorted	unsorted	sorted
Search(A, k)	$O(n)$	$O(\log n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Insert(A, x)	$O(1)$	$O(n)$	$O(1)$	$O(n)$	$O(1)$	$O(n)$
Delete(A, x) or Delete(A,k) (given location of x)	$O(1)^*$	$O(n)$	O(1) *	O(1) *	$O(1)$	$O(1)$
Remove(A,x) or Remove(A,k) (not given location of x)	O(n) requires search + delete	O(n)	O(n)	O(n)	O(n)	O(n)

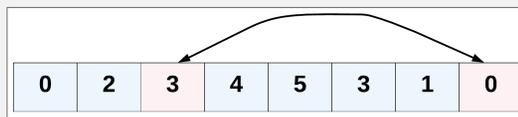A is the dictionary, k is a key, x is a key-value pair (k,v)

delete operation as defined in ADM assumes that the element is already found (known array index, pointer to the linked list node) – otherwise find operation is required first
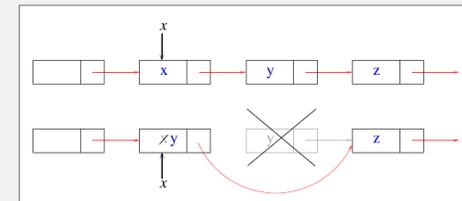
* denotes cleverness or subtlety

Constant-Time Deletion in an Unsorted Array

0	2	3	4	5	3	1	0

- O(1) deletion
  - swap element to be deleted with the last element, then remove the (new) last element

---

Constant-Time Deletion in a Singly-Linked List

- O(1) deletion

```
x.setValue(x.getNext().getValue())
x.setNext(x.getNext().getNext())
```