Universidad Carlos III de Madrid

Algorithms and Data Structures (ADS)

Bachelor in Informatics Engineering
Computer Science Department

YEAR: 1º / SEMESTER: 2º

Exam 2

Author:
Nathan D. Ryan

Junio 2011

Question 1
[10 points]

a)Provide a definition for the term *algorithm*.
b)Provide a definition for the term *data structure*.
a)What are the three things that define an ADT (abstract data type)?

Question 2
[20 points]

What is the asymptotic performance of the following operations of each data structure?

	Data Structure					
Operation	Singly Linked List	Doubly Linked List	Binary Search Tree	Balanced Binary Search Tree	Hash Table (best)	Hash Table (worst)
isEmpty()						
size()						
getFirst()						
getLast()						
get(K)						
setFirst(V)						
setLast(V)						
set(K, V)						
insertFirst(V)						
insertFirstAfter(K, V)						
insertLast(V)						
insertLastBefore(K, V)						
insert(K, V)						
removeFirst()						
removeLast()						
remove(K)						
clear()						

For lists, the key is the index of an element (a value of type `int`), and the value is the element.

Justify your answers for the `clear` *operation.*

Question 3
[30 points]

The "next" node of the last node of a singly-linked list is `null`. Suppose we set the "next" node of the last node to be the first node, effectively turning the chain of nodes into a circular chain of nodes. Further, instead of storing a reference to the first element, our singly-linked list will store a reference to the *last* element. Which of the operations of our circular singly-linked list still have an asymptotic performance of O(*n*)? Can anything be done to improve the asymptotic performance of those methods to O(1)?

Question 4
[10 points]

What is the risk of using recursion? When is it appropriate to use recursion?

Question 5
[20 points]

True/False. If a statement is false, describe why it is false.

a)The error conditions for the operations of a list are the same as the error conditions for the corresponding operations of a BST.

b)The keys of a BST must have a total ordering.

c)A good hash function evenly distributes values.

d)Every tree is a graph.

e)Using recursion to implement the methods of a balanced BST is appropriate.

f) Using recursion to implement the methods of a BST is appropriate.

g)Every tree is a list.

h) The height of a binary search tree determines the performance of the operations of the tree.

i) These aren't the droids you're looking for.

a) A *k*-ary tree (that is, a tree in which each node can have *k* children) can be represented as a binary tree, even if *k* is greater than 2.

Question 6
[30 points]

Consider the following interface for an undirected graph:

```
public interface Graph<N, E> {
   /**
    * Gets the nodes of this graph.
    */
   Set<N> getNodes();
   /**
    * Gets the edge between two nodes, or null if no such
edge
    * exists.
    */
   E getEdge(N node1, N node2);
}
```

A set is iterable:

```
public interface Set<E> extends Iterable<E> {
   ...
   /**
    * Gets an iterator over the elements of this set.
    */
   Iterator<E> iterator();
   ...
}
```

An iterator makes a pass over each element of a collection in sequence by successive calls of its `next()` method: