Universidad Carlos III de Madrid

Algorithms and Data Structures (ADS)

Bachelor in Informatics Engineering
Computer Science Department

YEAR: 1º / SEMESTER: 2º

Exam 1

Author:
Nathan D. Ryan

Junio 2011

C2.218.13873-89
2011-03-28T12:30:00+0200

Question 1 [10 points]

a)Provide a definition for the term *algorithm*.
b)Provide a definition for the term *data structure*.

Question 2 [20 points]

a)What are the three things that define an ADT (abstract data type)?
a)What additionally defines an ADS (abstract data store)?
b)A data type is a specific implementation of an ADT. Give an example (by name) of an ADT and a corresponding data type.
c)A data store is a specific implementation of an ADS. Give an example (by name) of an ADS and a corresponding data store.

Consider the following (partial) interface definition and corresponding implementation of a stack:

```
public interface Stack<E> {
    boolean isEmpty();
    ...
}

import java.util.ArrayList;
import java.util.List;
/** A stack backed by a linked chain of nodes. */
public class LinkedStack<E> implements Stack<E> {
    /** The list wrapped by this stack. */
    private List<E> elements = new ArrayList<E>();
    public boolean isEmpty() {
        return this.elements.isEmpty();
    }
    ...
}
```

a)Is `Stack` an ADS or only an ADT? Explain.
b)Is `LinkedStack` an ADS or only an ADT? Explain.

Question 3 [20 points]

Complete the following table by providing the ideal asymptotic performance of each operation for each implementation.

	LinearDataStore backed by:			
Operation	array	circular array	singly-linked nodes	doubly-linked nodes
isEmpty()				
getSize()				
getFirst()				
getLast()				
get(int)				
setFirst(E)				
setLast(E)				
set(int, E)				
insertFirst(E)				
insertFirstAfter(int, E)				
insertLast(E)				
insertLastBefore(int, E)				
removeFirst()				
removeLast()				
remove(int)				

Is the performance profile you provided for each implementation the only possibility? That is, for at least one entry of the table above, argue why the operation of the implementation might have a different asymptotic performance than what you listed.

Question 4 [20 points]

For each function below, determine the asymptotic bounding function (that is, the big-O for the function). Then, place the distinct asymptotic bounding functions in order with respect to one another, from slowest growing to fastest growing.

$2^n + 3^{n+1}$

6561

$\log_2 n$

$\log_2 n^2 + 1$

n^n

$\log_3 n$

$n^3 + 2n^2 - 2n + 1$

$12n^{1/2}$

Question 5 [30 points]

Consider the provided (partial) interface definition for a linear data store and the corresponding implementation. Suppose we want to add an additional operation to `LinearDataStore`:

```
/**
 * Inserts each element of the given data store at the end
of
 * this data store. The elements are added in the same
order.
 */
void insertLast(ArrayLinearDataStore<E> moreElements);
```

Assuming `moreElements` has *n* elements, write an implementation for this method in terms of existing methods (listed in the above table) such that the implementation has an amortized asymptotic performance of O(*n*):

```
public void insertLast(ArrayLinearDataStore<E> moreElements)
{
    // TODO implement
    ...
}
```

What would be the performance of your implementation if the type of `moreElements` was instead `SinglyLinkedLinearDataStore`?