

Curso de Estructuras de Datos y Algoritmos – 2008

Práctico 7

- 1.a. Escribir un módulo de especificación para el TAD **Stack** (Pila) de naturales, conteniendo un conjunto mínimo de procedimientos constructores, de predicado y selectores. La especificación debe ser funcional.
- b. Escribir un módulo de implementación para el TAD Stack de naturales. Utilice como representación una lista encadenada simple de punteros.

2. Suponga ahora que se utilizará una variante de Stack que posee una cota máxima para la cantidad de elementos.
 - a. Modificar la especificación del TAD Stack del ejercicio **1.a** para poder manejar Stack con cota máxima de elementos.
 - b. Escribir un módulo de implementación para el TAD Stack de naturales estático. Utilice como representación un arreglo de **n** posiciones.
 - c. Con un solo arreglo es posible implementar 2 de estos Stack. Dar una descripción de cómo serían las operaciones para manejarlos.

3. Usualmente los editores de texto incluyen algún carácter especial (por ejemplo "backspace") que tiene la funcionalidad de borrar el carácter previo no borrado. Por ejemplo, si # es este carácter, entonces el string *abc#d##e* es en realidad *ae*. Los editores de texto también incluyen, por lo general, un carácter de borrado de línea. Dicho carácter permite eliminar todos los caracteres previos de la línea de texto. A este carácter lo denotaremos con el símbolo @. Escribir un programa C/C++ que lea y luego imprima una línea de caracteres que puede contener ocurrencias de los caracteres de borrado arriba descritos.

- 4.a. Proponer una estructura de datos que soporte, además de las operaciones básicas del TAD Stack de naturales, una operación **Mínimo** que retorna el mínimo elemento de la estructura, teniendo en cuenta que el tiempo de ejecución de la operación Mínimo deber ser independiente de la cantidad de elementos del Stack.
- b. Dar una especificación e implementar el TAD StackMin.

5. Dada una expresión en notación infija, compuesta de operadores binarios + y *, números naturales y paréntesis, se pide:
 - a. Escribir un procedimiento que transforme la expresión a su representación en notación posfija. Para realizar esta operación Ud. debe usar un Stack de caracteres en donde ir colocando los operadores (según su orden de precedencia). La expresión de entrada es un arreglo de caracteres. La expresión de salida será una lista de operadores y operandos representando la expresión posfija. Asuma que las expresiones de entrada son sintácticamente correctas.
Ej.:
Expresión infija de entrada: $10*5+3$
Expresión posfija equivalente: $10\ 5\ *\ 3\ +$
Expresión infija de entrada: $10*(5+3)$

Expresión posfija equivalente: $10\ 5\ 3\ +\ *$
Expresión infija de entrada: $10+5*3$
Expresión posfija equivalente: $10\ 5\ 3\ *\ +$
Expresión infija de entrada: $(10+5)*3$
Expresión posfija equivalente: $10\ 5\ +\ 3\ *$
Expresión infija de entrada: $(10+5*10+3)*3$
Expresión posfija equivalente: $10\ 5\ 10\ *\ 3\ +\ +\ 3\ *$

b. Evaluar la expresión en notación posfija. Para realizar esta operación Ud. debe usar un Snack donde ir colocando los operandos. Se escribirá a tales efectos un procedimiento que recibirá la expresión a evaluar (lista de operadores y operandos) y retornará un número natural. Utilizar una estructura de datos adecuada para la lista de operadores y operandos.

6.

a. Escribir una especificación del TAD **Queue** (Cola). La especificación debe ser funcional y tiene

que tener un conjunto mínimo de constructores, predicados y selectores.

b. Escribir un módulo de implementación, dando una representación adecuada.

7. Consideraremos una variante del TAD Queue en la cual la memoria es acotada pero siempre es posible realizar inserciones. Si la Queue está llena y se realiza una inserción, se elimina de la memoria el elemento más viejo y se agrega el elemento a insertar como el más nuevo.

a. Modifique el TAD especificado en **6.a** para poder manejar este tipo de Cola.

b. Escriba el correspondiente módulo de implementación. Utilice como representación un arreglo y dos cursores que determinan el principio y fin de la Cola.

8. Considere el Tipo Abstracto de Datos Colas de Prioridad (Priority Queue). A diferencia del Tipo Abstracto Queue las inserciones dependerán de un parámetro agregado (la prioridad) y las bajas dependen de la prioridad de las tareas dentro de la cola.

Es importante notar que la prioridad puede tener una dependencia directa del tipo de dato a insertar. En ese caso tiene que existir una función de orden asociada al tipo de dato que permita realizar la clasificación de los elementos. Sin embargo, a los fines de este ejercicio consideraremos la prioridad como un dato externo al tipo a insertar. Adicionalmente asumiremos que un elemento tiene mayor prioridad que otro si es ingresado con un valor más bajo. Para representar las prioridades utilizaremos el tipo primitivo **SHORT**.

a. Escribir un módulo de especificación para el TAD Cola de Prioridad de n niveles. Esta especificación debe contener un conjunto mínimo funcional de constructores, predicados y selectores.

b. Escribir un módulo de implementación para el TAD Cola de Prioridad de n niveles.

9. Una **Deque** es una cola con dos extremos, donde las inserciones y las bajas pueden efectuarse en cualquiera de ellos. Se pide:

a. Escribir un módulo de definición del TAD Deque de naturales que incluya procedimientos para:

i. Crear una Deque vacía.

ii. Realizar las inserciones y bajas permitidas.

iii. Obtener los elementos situados en los extremos.

iv. Determinar si una deque es vacía o no.

b. Escribir un módulo de implementación, de forma que:

i. La representación de la deque sea una estructura de datos dinámica.

ii. Las implementaciones de los procedimientos insuman el mismo tiempo de ejecución para deque de cualquier largo.