

# TAREA #8 REPRESENTACIÓN EN MEMORIA DE LAS PILAS Y OPERACIONES BÁSICAS DE LAS PILAS

CINTHIA GUADALUPE OLIVAS CALDERON NO.17212165

## Representación en memoria de las pilas

Las pilas no son estructuras de datos fundamentales, es decir, no están definidas como tales en los lenguajes de programación. Las pilas pueden representarse mediante el uso de:

- Arreglos.
- Listas enlazadas.

Es importante definir el tamaño máximo de la pila, así como una variable auxiliar que se denomina TOPE. Esta variable se utiliza para indicar el último elemento que se insertó en la pila.

Al utilizar arreglos para implementar pilas se tiene la limitación de que se debe reservar el espacio en memoria con anticipación. Una vez dado un máximo de capacidad a la pila no es posible insertar un número de elementos mayor que el máximo establecido. Si esto ocurre, en otras palabras, si la pila está llena y se intenta insertar un nuevo elemento, se producirá un error conocido como desbordamiento –overflow

Una posible solución a este tipo de inconvenientes consiste en definir pilas de gran tamaño, pero esto resultará ineficiente y costoso si solo se utilizarán algunos elementos. No siempre es viable saber con exactitud el número de elementos a tratar, y siempre existe la posibilidad de que ocurra un error de desbordamiento.

Otro error que se puede presentar es tratar de eliminar un elemento de una pila vacía. Este tipo de error se le conoce como subdesbordamiento –underflow-

## Operaciones básicas de las pilas

Una pila cuenta con 2 operaciones imprescindibles: apilar y desapilar, a las que en las implementaciones modernas de las pilas se suelen añadir más de uso habitual.

Crear: se crea la pila vacía.

Apilar: se añade un elemento a la pila.(push)

Desapilar: se elimina el elemento frontal de la pila.(pop)

Cima: devuelve el elemento que está en la cima de la pila. (top o peek)

Vacía: devuelve cierto si la pila está vacía o falso en caso contrario

- Pila Vacía.

Es una operación auxiliar para verificar que si una PILA está vacía para poder insertar un nuevo elemento.

Este algoritmo verifica si una estructura tipo pila está vacía, asignando a BAND el valor de verdad correspondiente. La pila se implementa en un arreglo unidimensional. TOPE es un parámetro de tipo entero. BAND es un parámetro de tipo booleano

*Pila\_vacía (PILA, TOPE, BAND)*

*Si (TOPE = 0) (Verifica si no hay elementos almacenados en la pila)*

*Entonces*

*Hacer BAND VERDADERO (La pila está vacía)*

*Si no*

*Hacer BAND FALSO (La pila no está vacía)*

*Fin Si*

- Pila Llena

Es una operación auxiliar para validar que una pila no se encuentre llena antes de insertar un valor y/o para validar que no se dé el error de desbordamiento.

Este algoritmo verifica si una estructura tipo pila está llena, asignando a BAND el valor de verdad correspondiente. La pila se implementa en un arreglo unidimensional de MAX elementos. TOPE es un parámetro de tipo entero. BAND es un parámetro de tipo booleano.

*Pila\_llena (PILA, TOPE, MAX, BAND)*

*Si (TOPE = MAX)*

*Entonces*

*Hacer BAND VERDADERO (La pila está llena)*

*Si no*

*Hacer BAND FALSO (La pila no está llena)*

*Fin Si*

- Poner elemento (Push)

Este algoritmo agrega el elemento DATO en una estructura tipo – Pila –, y si la misma no está llena. Actualiza el valor TOPE. MAX representa el número máximo de elementos que puede alcanzar PILA. TOPE es un parámetro de tipo entero

*Push (PILA, TOPE, MAX, DATO)*

*Llamar a Pila\_Llena (PILA, TOPE, MAX, BAND)*

*Si (BAND VERDADERO)*

*Entonces*

*Escribir “Desbordamiento – Pila Llena”*

*Si no*

*Hacer TOPE <- TOPE + 1 y PILA [TOPE] DATO*

*(Actualiza TOPE e inserta el nuevo elemento en el TOPE de PILA)*

*Fin Si*

- Quitar elemento (Pop)

Este algoritmo saca un elemento – Dato – de una estructura tipo pila – PILA –, si ésta no se encuentra vacía. El elemento que se elimina es que se encuentra en la posición indicada por TOPE

*Llamar a Pila\_vacía (PILA, TOPE, BAND)*

*Si (BAND VERDADERO)*

*Entonces*

*Escribir “Subdesbordamiento – Pila vacía”*

*Si no*

*Hacer DATO PILA [TOPE] y TOPE -1 (Actualización de TOPE)*

*Fin Si*

Referencias

Joyanes Aguilar, L., & Zahonero Martínez, I. (1998). Estructura de datos. Algoritmos, abstracción y objetos. (Ed. rev.). Madrid, España: McGraw-Hill.

José Fager W. Libardo Pantoja Yépez Marisol Villacrés Luz Andrea Páez

Martínez Daniel Ochoa Ernesto Cuadros-Vargas. (2014). Tipos de Datos

Abstractos.En Estructuras de datos(222). Latinoamérica: LATIn.