

PILAS

Una pila (stack) es un tipo especial de lista lineal en la que la inserción y borrado de nuevos elementos se realiza sólo por un extremo que se denomina cima o tope (top). La pila es una estructura con numerosas analogías en la vida real: una pila de platos, una pila de monedas, una pila de cajas de zapatos, una pila de camisas, una pila de bandejas, etc.

Dado que las operaciones de insertar y eliminar se realizan por un solo extremo (el superior), los elementos sólo pueden eliminarse en orden inverso al que se insertan en la pila. El último elemento que se pone en la pila es el primero que se puede sacar; por ello, a estas estructuras se les conoce por el nombre de LIFO (last-in, first-out, último en entrar, primero en salir). Las operaciones más usuales asociadas a las pilas son: "push" Meter, poner o apilar: operación de insertar un elemento en la pila. "pop" Sacar, quitar o desapilar: operación de eliminar un elemento de la pila.

Las pilas se pueden representar en cualquiera de las tres formas:

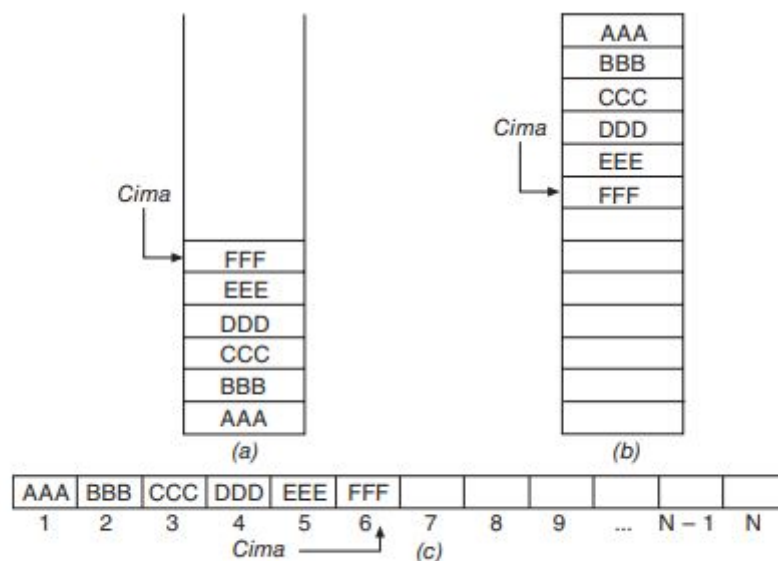
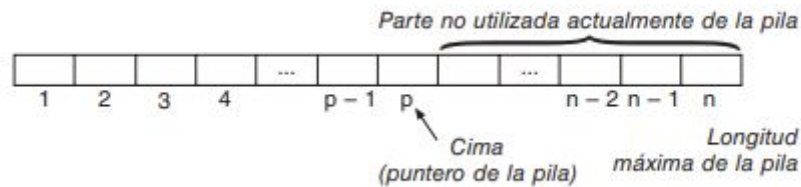


Figura 12.16. Representación de las pilas.

Para representar una pila St, se debe definir un vector con un determinado tamaño (longitud máxima):

var array [1..n] de : St

Se considerará un elemento entero P como el puntero de la pila (stack pointer). P es el subíndice del array correspondiente al elemento cima de la pila (esto es, el que ocupa la última posición). Si la pila está vacía, $P = 0$. (Véase la figura de la página siguiente.) En principio, la pila está vacía y el puntero de la pila o CIMA está a cero. Al meter un elemento en la pila, se incrementa el puntero en una unidad. Al sacar un elemento de la pila se decrementa en una unidad el puntero. Al manipular una pila se deben realizar algunas comprobaciones. En una pila vacía no se pueden sacar datos ($P = 0$). Si la pila se implementa con un array de tamaño fijo, se puede llenar cuando $P = n$ (n, longitud total de la pila) y el intento de introducir más elementos en la pila producirá un desbordamiento de la pila.



Idealmente una pila puede contener un número ilimitado de elementos y no producir nunca desbordamiento. En la práctica, sin embargo, el espacio de almacenamiento disponible es finito. La codificación de una pila requiere un cierto equilibrio, ya que si la longitud máxima de la pila es demasiado grande se gasta mucha memoria, mientras que un valor pequeño de la longitud máxima producirá desbordamientos frecuentes. Para trabajar fácilmente con pilas es conveniente diseñar subprogramas de poner (push) y quitar (pop) elementos. También es necesario con frecuencia comprobar si la pila está vacía; esto puede conseguirse con una variable o función booleana VACIA, de modo que cuando su valor sea verdadero la pila está vacía y falso en caso contrario.

P = CIMA Puntero de la pila.

VACIA Función booleana «pila vacía».

PUSH Subprograma para añadir, poner o insertar elementos.

POP Subprograma para eliminar o quitar elementos.

LONGMAX Longitud máxima de la pila.

X Elemento a añadir/quitar de la pila.

Bibliografía:

Luis Joyanes Aguilar, Ignacio Zahonero Martinez. (2008). Estructura de datos en Java. Mexico: MCGRAW-HILL.