

Unidad 2

Pilas

Estructura de Datos

Definición de pila

Una pila es una lista que permite almacenar y recuperar datos. Se caracteriza, primordialmente, por el modo en que accede a sus elementos, representado en dos operaciones básicas: adicionar un elemento al final de la pila y eliminar un elemento desde el final de la misma; es decir, los elementos se agregan o eliminan desde la última posición, lo cual se identifica como LIFO (*Last In First Out*), dicho de otra manera: el último elemento en entrar, es el primero en salir (Proyecto Latín, 2014).

En la pila solo se pueden insertar y eliminar nodos en uno de los extremos de la lista. Dichas operaciones se reconocen como push y pop, correspondientemente apilar (insertar) y desapilar (borrar). Una pila se puede representar como en la figura.

Todas las operaciones trabajan sobre un único elemento que se puede alcanzar en la estructura: el último que se apiló. Podemos concluir, entonces, que los elementos se sacan de la estructura en el orden inverso al que se han insertado (Sáez, 2011).

Estructura de una pila

Una pila puede estructurarse como un arreglo de datos o una lista enlazada. En el primer caso, se debe indicar la cantidad de elementos totales que se van a tener en total en la lista. En el segundo caso, se requiere un conjunto de nodos que permitan almacenar un dato y la posición o puntero al siguiente elemento o nodo en la pila (Juganaru, 2014).

Observa con atención el video ***Pilas en C++ Concepto y Ejemplo***.

Enlace del video: <https://www.youtube.com/watch?v=6B99QKHflFA&feature=youtu.be>

En él podrás ver que, si se utilizan arreglos para crear la lista, se tiene un inconveniente: la pila será estática, no puede aumentar el total de elementos a guardar; además, si no se usan todas las posiciones disponibles, se desperdicia espacio en memoria.



Cuando se usan nodos, el tamaño de la pila es dinámico.

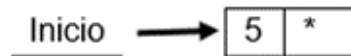
Para el caso de los nodos, se puede visualizar como una lista simple. Cada nodo está formado por dos atributos y campos: información o dato, y apuntador al siguiente nodo de la lista. Adicionalmente, la pila contiene un atributo que indica la posición inicial; si está vacía es nulo, sino, apunta al último nodo.

En primera instancia, la pila arranca sin datos, entonces el inicio apunto a nulo (*):



En este caso la pila está vacía.

Si se inserta un nuevo dato, como el número 5, se crea un nuevo nodo, al cual el inicio hará referencia.



Si nuevamente se inserta un dato, como el número 9, se añade un nuevo nodo al inicio y dicho nodo se vincula al anterior existente.



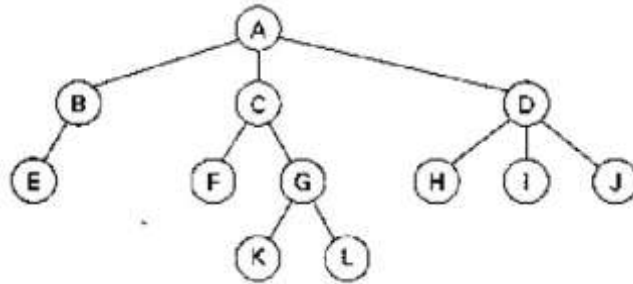
Actividad - Ejercicios de práctica 2

Teniendo en cuenta lo estudiado hasta el momento en la Unidad 2, realiza la implementación para los siguientes casos:

- Crea una clase en JAVA que permita definir una pila construida a partir de un arreglo.
- Adiciona un método que permita mostrar cuántos datos se pueden agregar a la pila o si ya está llena.
- Crea un objeto de esta clase con un tamaño de nueve (9).
- Apila cinco valores numéricos aleatorios.
- Muestra el tamaño actual de la pila.
- Borra un dato.
- Indica nuevamente el tamaño actual.
- Enseña la cantidad de datos que se pueden agregar a la pila.
- En JAVA, utiliza la clase Stack para crear dos nuevas pilas. Agregar siete (7) datos a la primera pila. La segunda pila deberá contener los mismos datos de la primera, pero en orden inverso.

Definición de árboles

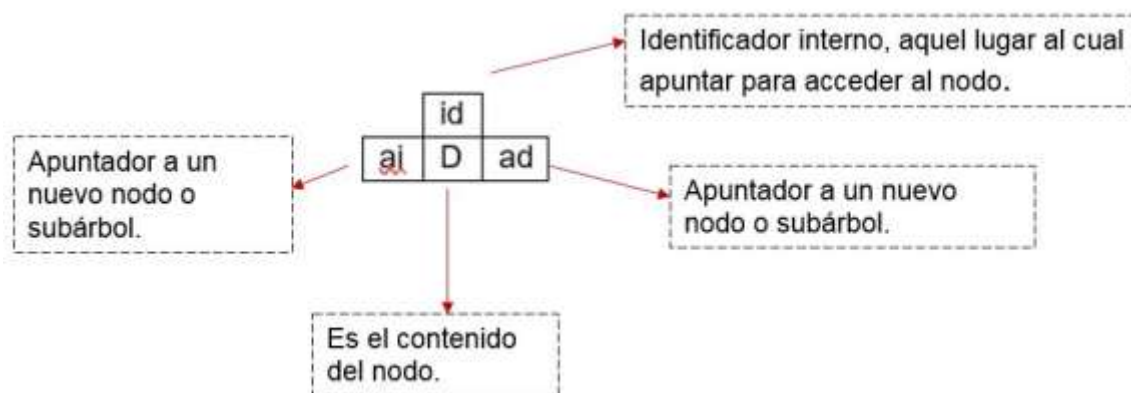
Un árbol es una estructura que consiste en diferentes nodos que pueden apuntar a uno o varios nodos. Su estructura es jerárquica, donde el primer elemento es llamado nodo raíz y permite que cada elemento tenga más de un sucesor (opuesto a lo que pasa en la lista, en la cual cada elemento tiene como máximo, uno) (Gutiérrez, 2004).



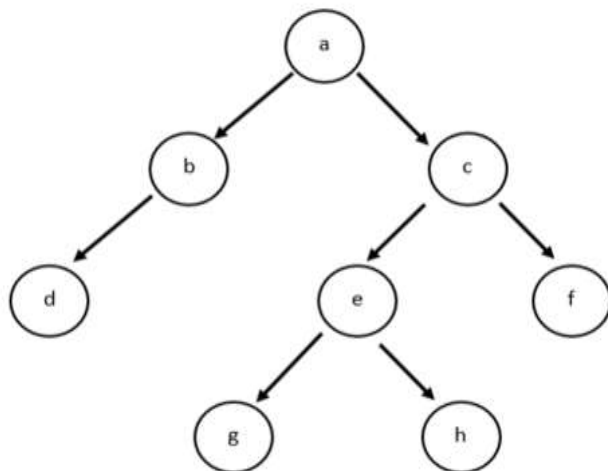
A los nodos al interior del árbol también se les denomina subárbol, y es una estructura recursiva, es decir, sus componentes están basados en su propia definición; de la misma forma ocurre con el código, se trata de una función que llama a la misma función: es una construcción a partir de sí misma.

Estructura de los árboles

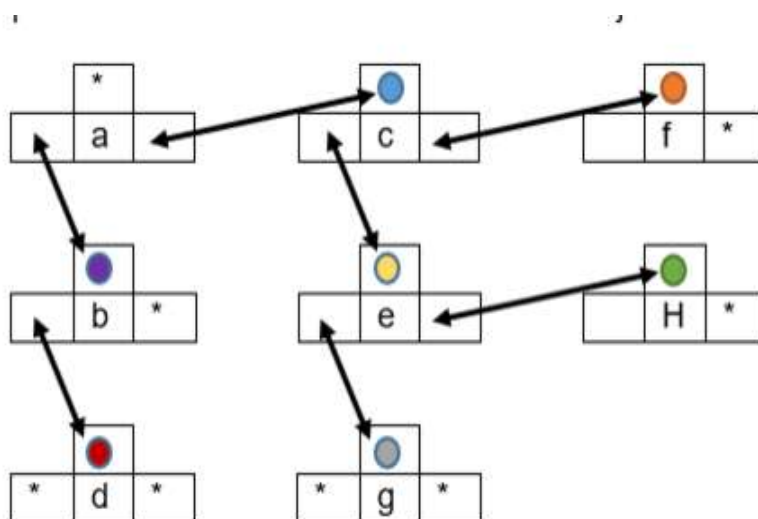
Los árboles se basan en nodos y su estructura se puede visualizar de la siguiente forma:



A continuación, se presentan dos figuras. En la primera se muestra la estructura de un árbol binario, en la cual los apuntadores pueden tener como máximo dos nodos hijos. En la segunda imagen, encontrarás la representación del árbol.



Esta estructura o árbol se puede representar como:



Adicionalmente, el árbol se puede representar con la siguiente agrupación:

(a (b (d), c (e (g, h), f)))

Glosario

A

Algoritmo

Conjunto ordenado de operaciones sistemáticas que permite hacer un cálculo y hallar la solución de un tipo de problemas.

Arreglos

Zona de almacenamiento contiguo que contiene una serie de elementos del mismo tipo.

C

Ciclos

Sentencia que ejecuta repetidas veces un trozo de código, hasta que la condición asignada a dicho bucle, deja de cumplirse.

Clase

Plantilla para la creación de objetos de datos según un modelo predefinido.

D

Datos simples

Representan un solo valor.

E

Estructura de datos

Forma particular de organizar datos en una computadora, para que puedan ser utilizados de manera eficiente.

J

JAVA

Lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

M

Métodos

Subrutina cuyo código es definido en una clase y puede pertenecer tanto a una clase como a un objeto.

O

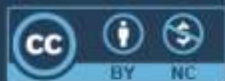
Objeto

Unidad dentro de un programa de computadores que consta de un estado y de un comportamiento, que, a su vez, constan respectivamente de datos almacenados y de tareas realizables durante el tiempo de ejecución.

Bibliografía

- Gutiérrez, F. (2004). *Estructuras de datos: especificación, diseño e implementación*. Ediciones UPC.
- Joyanes, A. & Zahonero, M. (2005). Programación en C: metodología, algoritmos y estructura de datos (2ª. ed.).
<http://ebookcentral.proquest.com/lib/iudasp/detail.action?docID=3195036>
- Joyanes, A. & Zahonero, M. (2007). *Estructura de datos en C++*. McGraw-Hill.
<http://ebookcentral.proquest.com/lib/iudasp/detail.action?docID=3194764>
- Juganaru, M. (2014). *Introducción a la Programación*. Grupo Editorial Patria.
<http://ebookcentral.proquest.com/lib/iudasp/detail.action?docID=3227908>
- Proyecto Latín. (2014). *Estructuras de Datos*. LaTIn.
<https://openlibra.com/es/book/estructuras-de-datos>
- Sáez, X. (2011). *Prácticas de programación. Estructuras de datos básicas: secuencias*. Eureka Media.
- Vilanova, R. & Lagonigro, R. (2010). *Tipos estructurados de datos: Tablas y tuplas*. UOC. <https://openlibra.com/es/book/tipos-estructurados-de-datos-tablas-y-tuplas>

Esta licencia permite a otros distribuir, remezclar, retocar, y crear a partir de esta obra de manera no comercial y, a pesar que sus nuevas obras deben siempre mencionar a la IU Digital y mantenerse sin fines comerciales, no están obligados a licenciar obras derivadas bajo las mismas condiciones.



IUDigital
de Antioquia
INSTITUCIÓN UNIVERSITARIA
DIGITAL DE ANTIOQUIA