

Colas y Pilas

Algoritmos y Estructuras de datos

Prof. Ing. Gomez Pablo



24 de abril de 2022

Introducción

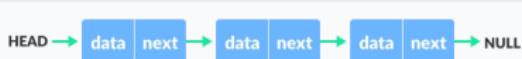
Palabras Clave

- Colas
- Listas
- FIFO/LIFO
- Especificación Algebraica
- Aridad de funciones
- Calculadora posfija
- Balance de paréntesis
- JSON

Introducción

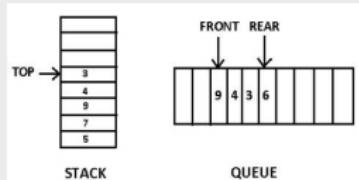
Lista

- Es una estructura de datos que almacena una cadena de datos mediante una referencia (puntero) al siguiente elemento.



Pila (Stack) y Cola (Queue)

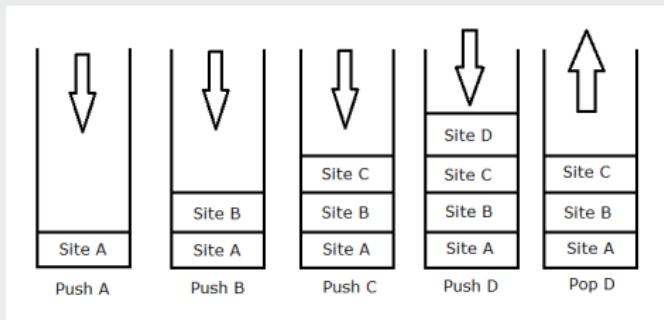
- Pueden ser implementadas usando arrays o listas.
- No permiten agregar o remover elementos en cualquier índice.
- Al igual que las listas son estructuras lineales contenedoras de datos. Hacemos referencia a una pila, cola de datos de tipo A.



Pila vs Cola

Pila

- Es una estructura LIFO (Last in first out)
- Ej: Historial de navegación. Se navega al Sitio A \Rightarrow B \Rightarrow C \Rightarrow D. A medida que el usuario navega se apilan los diferentes sitios en la lista de sitios. Esto asegura que el último sitio visita esté siempre en el tope de la pila. Cuando el usuario presiona el botón “Volver” se remueve el tope de la pila y se accede al sitio anterior C que quedará como tope de la pila.
- Undo/Redo



Pila vs Cola

Cola

- Es una estructura FIFO (First in first out)
- Ej: Fila del supermercado. Cuando el cajero está cobrando a los clientes atenderá al primero que llegó a la fila.
- Son utilizadas como buffers. Modelo de productor y consumidor.



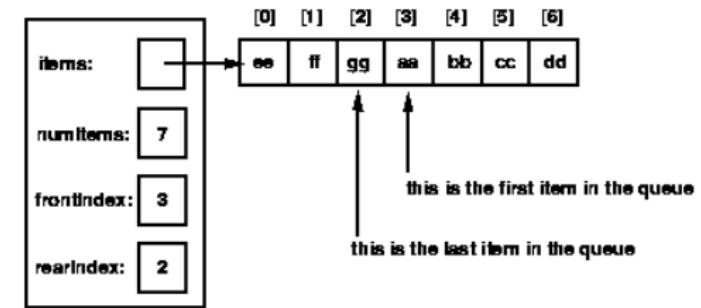
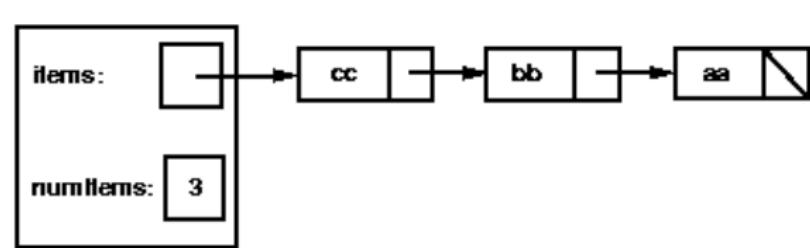
Pilas y Colas - Implementación

Debe respetarse el comportamiento de cada tipo FIFO/LIFO.



Pilas y Colas - Implementación

Puede utilizarse un arreglo o una lista (memoria estatica / dinamica)



Pilas - Especificación Algebraica

Aridad de funciones

- crearPila: \Rightarrow Pila *
- apilar:Pila \times A \Rightarrow Pila *
- esVacia:Pila \Rightarrow Boolean
- tope:Pila \Rightarrow A
- desapilar:Pila \Rightarrow Pila

* Funciones constructoras

Comportamiento P: Tipo Pila , a: Tipo A

- vacia(crearPila())=true
- vacia(apilar(P,a))=false
- tope(crearPila()) \Rightarrow error
- tope(apilar(P,a))=a
- desapilar(crearPila()) \Rightarrow error
- desapilar(apilar(P,a))=P

Colas - Especificación Algebraica

Aridad de funciones

- $\text{crearCola}: \Rightarrow \text{Cola}$, $\text{encolar}:\text{Cola} \times A \Rightarrow \text{Cola}^*$
- $\text{esVacia}:\text{Cola} \Rightarrow \text{Boolean}$, $\text{tope}:\text{Cola} \Rightarrow A$
- $\text{desencolar}:\text{Cola} \Rightarrow \text{Cola}$

* Funciones constructoras

Comportamiento C:Tipo Cola , a b: Tipo A

- $\text{vacia}(\text{crearCola}())=\text{true}$
- $\text{vacia}(\text{encolar}(C,a))=\text{false}$
- $\text{tope}(\text{crearCola}()) \Rightarrow \text{error}$
- $\text{tope}(\text{encolar}(\text{crearCola}(),a))= a$
- $\text{tope}(\text{encolar}((\text{encolar}(C,b)),a))= \text{tope}(C)$
- $\text{deselencolar}(\text{crearCola}()) \Rightarrow \text{error}$
- $\text{desencolar}(\text{encolar}(C,a)) = C \text{ si } C = \text{crearCola}()$
- $\text{desencolar}(\text{encolar}(\text{encolar}(C,b)),a)) = \text{encolar}(\text{desencolar}(C),a)$

Pilas y Colas - Implementación

Despliegue

C es una cola donde se encola primero 1 y luego 2

desencolar(encolar(encolar(createCola(),1),2))

encolar(desencolar(encolar(createCola(),1)),2)

encolar(createCola(),2)



Calculadora Posfija

Notaciones de expresiones algebraicas

Expresión algebraica

$$a - c * (d+e)$$

$$a - c * d + e \Rightarrow a - (c*d) + e$$

- Notación prefija - Polaca: operador se escribe primero y luego los operandos (+12) (*23)
- Notación infija : operador va entre medio de los dos operandos (1+2) (2*3)
- Notación posfija - Polaca inversa: operador va luego de los dos operandos (12+) (23*)

Calculadora Posfija

Calculadora Posfija

Infija vs Posfija

Infix Notation

$A + B * C \Rightarrow A + B * C \Rightarrow A + B C *$

$A + B C * \Rightarrow A + B C * \Rightarrow A B C * +$

Postfix Notation

Parentesis

Infix Expression

$(A + B) * C$

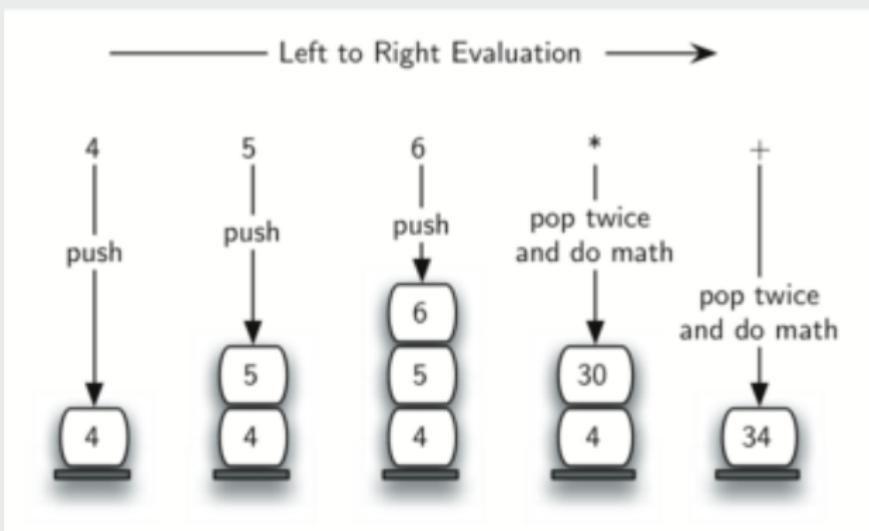
Postfix Expression

$A B + C *$

Calculadora Posfija

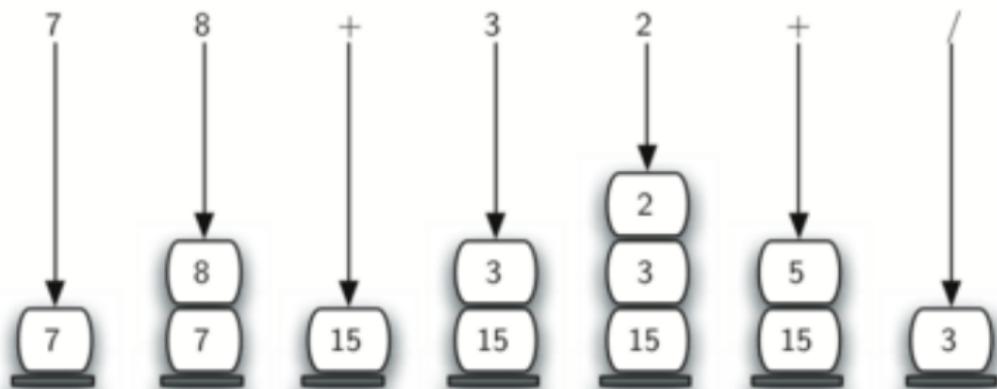
Evaluación

Tenemos la expresión posfija $4 \ 5 \ 6 \ * \ +$



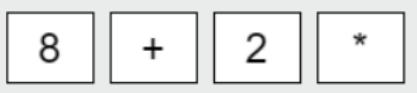
Calculadora Posfija

Evaluación



Calculadora Posfija Ejemplo 1

Tope  ⇒ Pila con tope en 3

Tope  ⇒ Desapilado [3]

Tope  ⇒ Desapilado [3,8]

Tope  ⇒ Desapilado [3,8,+]

Tope  ⇒ Se apila resultado 11

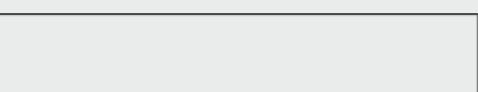
Calculadora Posfija Ejemplo 1

Tope  ⇒ Desapilado [11]

Tope  ⇒ Desapilado [11,2]

Tope  ⇒ Desapilado [11,2,*]

Tope  ⇒ Se apila resultado 22

Tope  ⇒ Pila vacía

El resultado es el último valor desapilado

Calculadora Posfija - Ejemplo 2

Calculadora Posfija



Ejercicio

Si tenemos la expresión :

$$2 * 3 + 3 - 2 * 1 = 6 + 3 - 2 = 7 \quad (1)$$

obtenemos en posfijo:

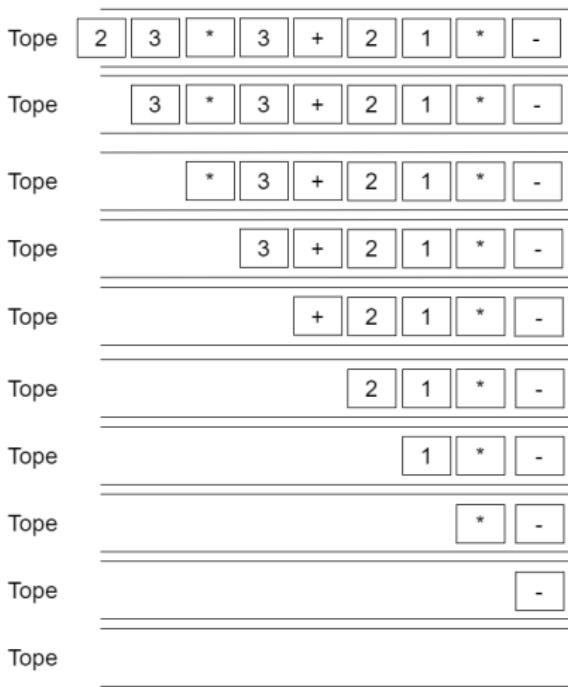
$$2\ 3\ *\ 3\ +\ 2\ 1\ *\ - \quad (2)$$

Realizar la conversión a posfijo de la siguiente expresión y su resolución:

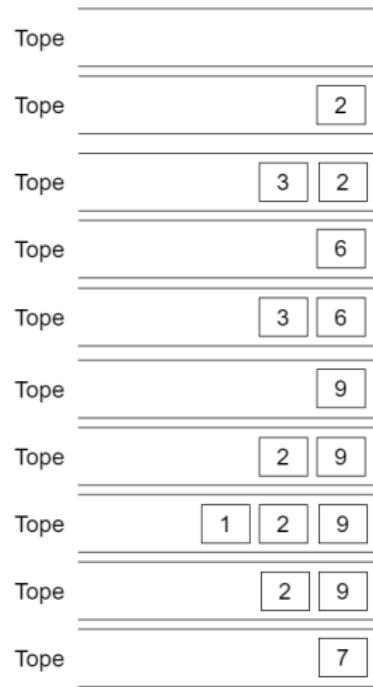
$$2 * ((3 + 3) - 2) * 1 = 2 * 4 * 1 = 8 \quad (3)$$

Calculadora Posfija - Modelo

Pila



Pila Auxiliar



Calculadora Posfija - Resolución

$$2 * ((3 + 3) - 2) * 1 \quad (1)$$

2 3 3 + 2 - * 1 *

	Pila	Pila Auxiliar
Tope	2 3 3 + 2 - * 1 *	
Tope	3 3 + 2 - * 1 *	2
Tope	3 + 2 - * 1 *	3 2
Tope	+ 2 - * 1 *	3 3 2
Tope	2 - * 1 *	6 2
Tope	- * 1 *	2 6 2
Tope	* 1 *	4 2
Tope	1 *	8
Tope	*	1 8
Tope		8

Calculadora Posfija - Especificación Algebraica

Aridad de funciones

["5", "1", "7", "*", "-", "10", "+"] Pila de Strings

- calculadora: Pila< string > ⇒ Int
- calcaux: Pila< string > x Pila< string > ⇒ Int
- esNum: string ⇒ boolean
- esOper: string ⇒ boolean
- computar: Pila x Operador ⇒ Pila
- suma: Int x Int ⇒ Int
- resta: Int x Int ⇒ Int
- multiplicacion: Int x Int ⇒ Int
- division: Int x Int ⇒ Int

Calculadora Posfija - Especificación Algebraica

Comportamiento

- $\text{calculadora}(P) = \text{calcaux}(P, \text{crearPila}())$
- $\text{calcaux}(P_p, P_a) =$
 - Si $\text{esVacia}(P_p) \Rightarrow \text{tope}(P_a)$
 - Si $\text{esNum}(\text{tope}(P_p)) \Rightarrow \text{calcaux}(\text{desapilar}(P_p), \text{apilar}(P_a, \text{tope}(P_p)))$
 - Si $\text{esOper}(\text{tope}(P_p)) \Rightarrow$
 $\text{calcaux}(\text{desapilar}(P_p), \text{computar}(P_a, \text{tope}(P_p)))$
- $\text{computar}(P_a, O) =$
 - Si $O = '+' \Rightarrow$
 $\text{apilar}(\text{desapilar}(\text{desapilar}(P_a)), \text{suma}(\text{tope}(P_a), \text{tope}(\text{desapl}(P_a))))$
 - Si $O = '-' \Rightarrow$
 $\text{apilar}(\text{desapilar}(\text{desapilar}(P_a)), \text{resta}(\text{tope}(P_a), \text{tope}(\text{desapl}(P_a))))$

P_p es la pila principal , P_a es la pila auxiliar

Balanceo de paréntesis

- Por cada paréntesis (apilo.
- Por cada paréntesis) desapilo.
- Si la pila es vacía al final , entonces la expresión está balanceada.

