



Algoritmos y Estructura de Datos

2024

Profesores:

Cueva, R. | Allasi, D. | Roncal, A. | Huamán, F.

0581

0582

0583

0584

Capítulo 2

PILAS



PILAS

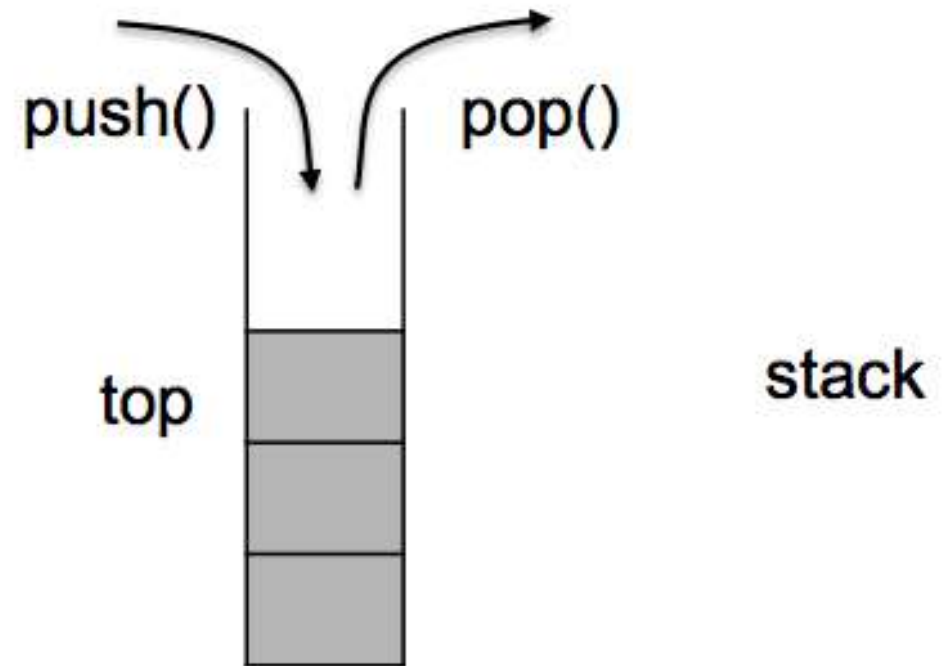
Las pilas son una colección de datos con las que nos familiarizamos desde la infancia. Desde la primera vez que comenzamos a apilar bloques uno encima del otro aprendemos que no es conveniente extraer los bloques de la parte inferior de la pila sin antes extraer los de la parte superior.

Ejemplo de pilas son: la pila de platos, las pilas de bandejas en una cafetería, etc.



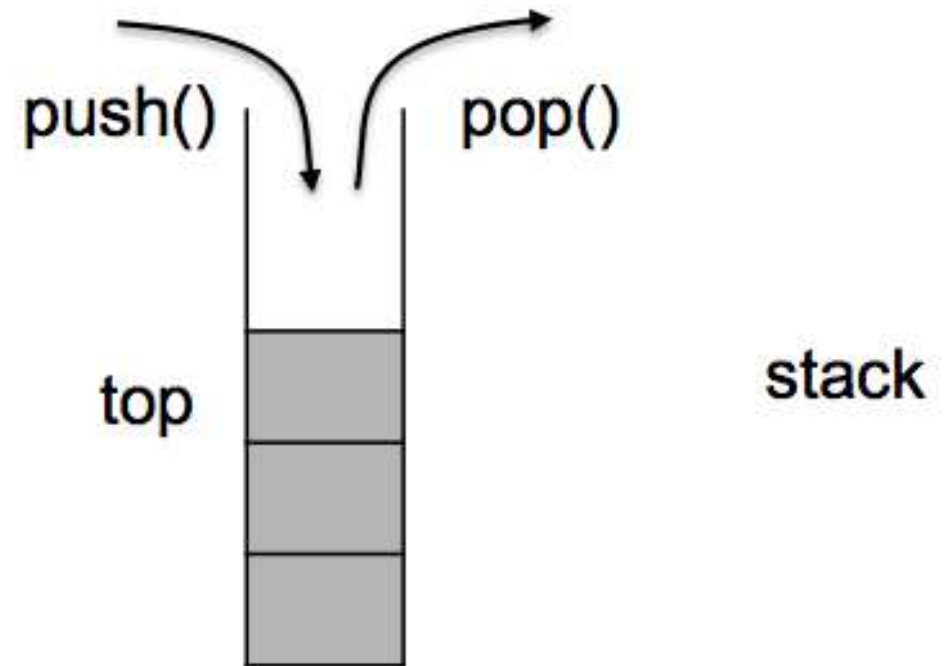
PILAS – Operaciones

- **Apilar()**: añade un elemento a la parte superior de la pila
- **Desapilar()**: elimina un elemento de la parte superior de la pila
- **Cima()**: examina el elemento situado en la parte superior de la pila
- **EsPilaVacía()**: determina si la pila está vacía
- **Tamaño()**: determina el número de elementos de la pila



PILAS – Operaciones

- **Apilar()**: añade un elemento a la parte superior de la pila
- **Desapilar()**: elimina un elemento de la parte superior de la pila
- **Cima()**: examina el elemento situado en la parte superior de la pila
- **EsPilaVacía()**: determina si la pila está vacía
- **Tamaño()**: determina el número de elementos de la pila



PILAS

EJERCICIOS



Cálculo en notación postfija

▪ Evaluación de expresiones aritméticas

- En la aritmética tradicional se necesita usar paréntesis para dar significado a las expresiones

$$A * B - C / D \qquad (A * B) - (C / D)$$

- Notación polaca (prefija), los operadores aparecen antes de los operandos, evitando el uso de paréntesis

$$- * AB / CD$$

- Notación polaca inversa (postfija), los operadores aparecen después de los operandos

$$AB * CD / -$$



Cálculo en notación postfija

- Cálculo de expresiones usando notación postfija (con una pila)
 - *Apilar los operandos hasta encontrar un operador*
 - *Desapilar los operandos, calcular y apilar el resultado*
 - *Hasta llegar al final de la expresión*
- Implementar una función que calcule el valor de una expresión postfija pasada como parámetro y usando el TAD Pila.



Cálculo en notación postfija - Algoritmo

Postfija (Cadena: expresion), retorna entero

Crear pila

Mientras haya caracteres en la expresión

X = siguiente símbolo en la expresión

Si x es operando entonces **push(P)**

Sino

Desapilar los operandos {dos **pop(P)**}

Calcular el resultado

Apilar el resultado {**push(P)**}

FinSi

FinMientras

Valor = **pop(P)**

Fin



Recorrer un laberinto

Suponga que construimos la matriz bidimensional de valores enteros, donde cada número representa una posición libre (1) o una pared (0) dentro de un laberinto.

Nuestro objetivo es comenzar en la esquina superior izquierda y recorrer el laberinto hasta alcanzar la esquina inferior derecha, recorriendo únicamente aquellas posiciones marcadas como libres.

1	0	1	1	1
1	1	1	0	1
0	0	1	0	1



Recorrer un laberinto

Los movimientos válidos serán todos aquellos que se realicen dentro de la cuadrícula y que tengan como destino una celda marcada por 1. A medida que recorremos el laberinto cambiamos 1 por 3 para indicar el camino seguido y **solo insertamos en la pila los movimientos válidos.**

Comenzando por la esquina superior izquierda solo tenemos **un movimiento válido**: hacia **abajo**.

3	0	1	1	1
1	1	1	0	1
0	0	1	0	1



Recorrer un laberinto

Insertamos el movimiento en la pila, lo **desapilamos** y **nos movemos a esa ubicación** eso quiere decir que nos hemos desplazado una posición hacia abajo.

Continuamos así apilando, desapilando y avanzando por todos aquellos movimientos válidos, en este caso hacia la derecha

3	0	1	1	1
3	3	3	0	1
0	0	1	0	1



Recorrer un laberinto

En este momento nos damos cuenta que vamos a tener dos movimientos válidos, **insertamos ambos movimientos**, extraemos de la pila la cima (movimiento hacia abajo) nos movemos hacia esa ubicación pero ya no tenemos movimientos válidos, pero la pila no está vacía, entonces, extraemos de la pila el anterior movimiento válido (**arriba**) y nos desplazamos hacia él, y continuamos el procesamiento. Al final, la ruta válida tiene 7.

3 0 1 1 1

3 3 3 0 1

0 0 3 0 1

7 0 7 7 7

7 7 7 0 7

0 0 3 0 7



PILAS - Resumen

- Las pilas operan sobre principios similares a las pilas del mundo real.
- Los elementos de una pila se procesan según un algoritmo LIFO: el último elemento en entrar es el primero en salir.
- Si necesitamos acceder a los elementos de la parte central o inferior, la pila no es la estructura apropiada para este problema.
- Debido a la relación existente entre las pilas y la recursión, siempre podemos reescribir el programa recursivo para transformarlo en un programa no recursivo que utilice una pila. En lugar de utilizar la recursión para ir almacenando los datos relevantes, podemos crear nuestra propia pila para hacerlo.



PILAS - Bibliografía

- Melgar, A. (2021). Estructuras de Datos (TAD) y sus implementaciones. [PowerPoint slides]. Facultad de Ciencias e Ingeniería. PUCP.
- Cueva, R. (2022, Agosto 21). Capítulo 2. Estructuras de Datos (TAD) y sus implementaciones. [PowerPoint slides]. Facultad de Ciencias e Ingeniería. PUCP.
- Roncal, A. (2023). Capítulo 5. Pilas. [PowerPoint slides]. Facultad de Ciencias e Ingeniería. PUCP.
- Hernández, R., Lázaro, J. C., Dormido, R. y Ros, S. (2001). Estructuras de Datos y Algoritmos. Pearson Education, S.A., Madrid
- Lewis, J. y Chase, J. (2006) Estructuras de datos con Java diseño de estructuras y algoritmos. Pearson Education S.A., Madrid

