



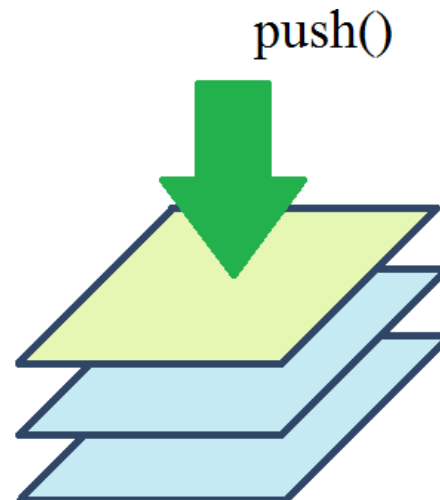
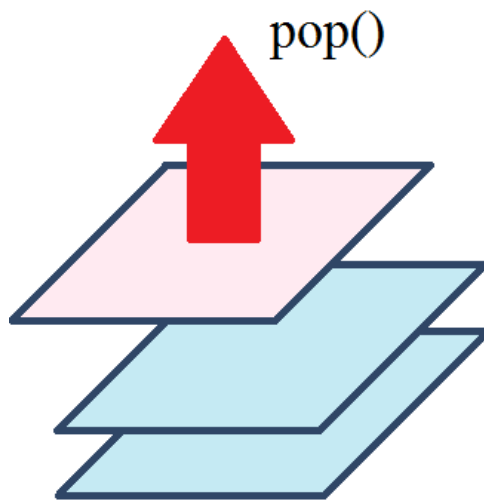
Universidad Nacional Autónoma de
México

Facultad de ingeniería



Actividad. Lunes #02: Algoritmo. Operación push y pop

Estructura de Datos y Algoritmos I



Alumno: González Medina Víctor Manuel

Profesor: M.I. Marco Antonio Martínez Quintana

Fecha: 16/06/2021

Semestre: 2021-2

Algoritmos push y pop

Como ya habíamos visto las pilas son una estructura de datos, y pues como ya sabemos el fin de estas, siempre es optimizar el programa que se esté desarrollando, y así tener un rendimiento mejor, y más que nada mayormente eficiente.

Las pilas ahora sí que como su nombre lo indica no son más que para apilar cosas, desde una pila de libros, de ropa, hasta lo que nos imaginamos, es una organización, en este caso es una organización de también elementos, pero los cuales conviven en la programación. Y algunas de las operaciones que se puede realizar con esta estructura son la operación push y la operación pop, que son de las cuales realizaremos sus algoritmos.

Algoritmo general. Operación push.

- Crear la pila
- Declarar el tipo de variable
- Iniciarla en 0 o el valor nulo
- Añadir elementos (operación *push*, declararla, para añadir elementos)
- Verificar si la pila está vacía
- Verificar si la pila esta completa
- Fin algoritmo

Algoritmo general. Operación pop

- Verificar si hay una pila vacía
- Verificar si hay elementos en una pila
- Quitar elementos (operación *pop*, declararla, para así llamar al elemento)
- Fin algoritmo

Algoritmo(pseint) de operación push y pop

Algoritmo operación push

Algoritmo Operacionpushpop

Dimensión pila[5]

max \leftarrow 0

Funcion añadir (elemento)

Si (No (pilaCompleta())) **Entonces**

max \leftarrow max + 1

pila[max] \leftarrow elemento

SiNo

Escribir "La pila esta llena, llego a su capacidad"

FinSi

FinFuncion

Algoritmo operación pop

Funcion remo \leftarrow quitar(pila[])

si (No(pilaVacia())) **Entonces**

remo \leftarrow pila[max]

max \leftarrow max-1

SiNo

Escribir "La pila esta vacía, no hay elementos"

remo \leftarrow 0

FinSi

FinFuncion

Conjunción de ambos algoritmos

```
Funcion max<- pilaVacia ()
```

```
    max= 0
```

```
FinFuncion
```

```
Funcion max<- pilaCompleta()
```

```
    max=5
```

```
FinFuncion
```

```
· pila_añadir(10)
```

```
· pila_añadir(19)
```

```
· pila_añadir(5)
```

```
·      Mientras (no( pilaVacia())) Hacer
```

```
·      Escribir pila quitar ()
```

```
      finMientras
```

```
· FinAlgoritmo
```