



**Universidad Nacional  
Autónoma de México**

**Facultad de Ingeniería**



**División de Ciencias Básicas**

**Estructura de Datos y  
Algoritmos I**

*Alumno: Bear Almaraz Miguel Ángel*

*Semestre 2021-2*

Nombre de la actividad:

***Actividad  
06(Miércoles):  
Pilas***

*Fecha: 09/06/2021*

## Pila

La pila (o stack) es una estructura de datos lineal y dinámica, en la cual el elemento obtenido a través de la operación ELIMINAR está predefinido, debido a que implementa la política Last-In, First-Out (LIFO), esto es, el último elemento que se agregó es el primer que se elimina

Las operaciones que se pueden realizar sobre una pila son INSERTAR (que es llamada PUSH) y ELIMINAR (que es llamada POP). Debido a la política LIFO que implementa esta estructura, el orden en el que los elementos son extraídos de la pila (POP) es inverso al orden en el que los elementos fueron insertados en la pila (PUSH). Además, el único elemento accesible de la pila es el que está hasta arriba y que se conoce como tope de la pila. Para poder diseñar un algoritmo que defina el comportamiento de una pila se deben considerar 3 casos para ambas operaciones (push y pop): ♣ Estructura vacía (caso extremo). ♣ Estructura llena (caso extremo). ♣ Estructura con elemento(s) (caso base).

## Operaciones

### Push

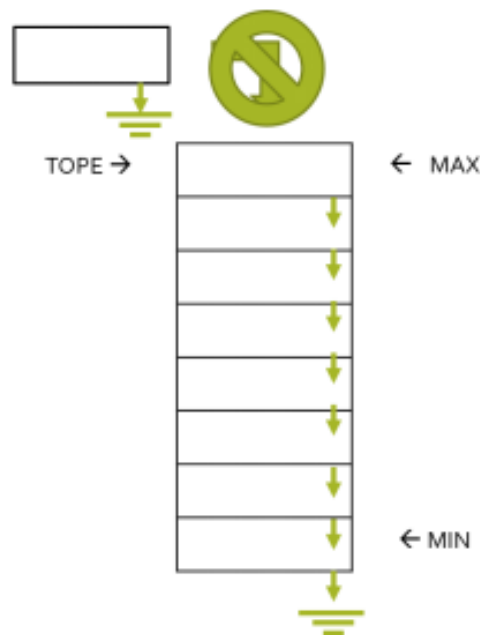
Esta operación sirve para insertar un elemento  $e$  en la pila  $S$ , lo vamos a escribir como:  $\text{push}(S, e)$ . Después de hacer esta operación sucede que: El elemento en la cima de la pila  $S$  ahora es  $e$ .



- (1) La operación push recibe : la dirección de una estructura pila y un elemento entero.
- (2) Incrementa el tope (cima) de la pila para agregar el elemento en una posición libre de la pila.
- (3) Asignando el valor  $e$  en la casilla  $S \rightarrow \text{top}$ .

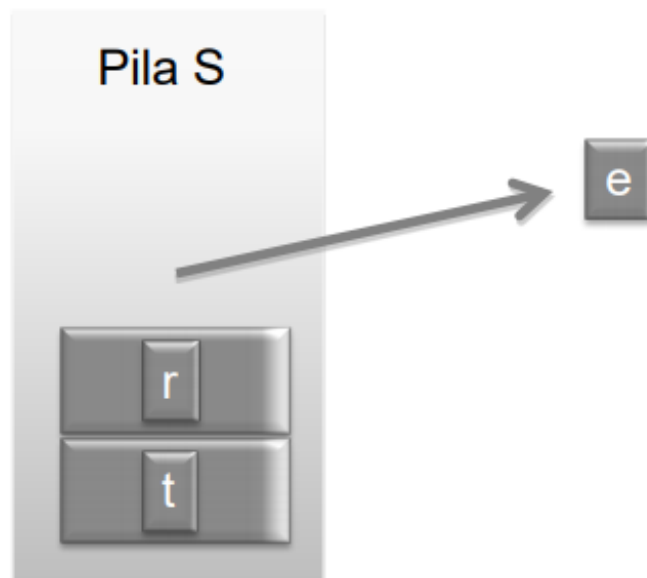
```
void push(Stack *S, int e) {  
    if(S->top == (STACKSIZE-1))  
        printf("Full stack\n");  
    else {  
        S->top++;  
        S->nodes[S->top] = e;  
    }  
}
```

En una pila llena no es posible hacer PUSH de un nuevo elemento, ya que se ha alcanzado el tamaño máximo permitido



### Pop

Esta operación sirve para retirar el elemento en la cima de la pila S, lo vamos a escribir como: pop(S,e)



La función devuelve un tipo entero al recibir la dirección de una variable de tipo estructura pila (struct stack \*). Las líneas (4) y (5) son las más importantes ya que se almacena el valor que será devuelto y se decrementa el tope de la pila.

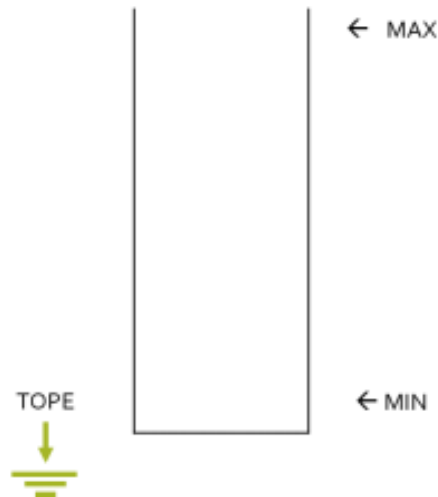
```
int pop(Stack *ps) {  
    if(ps->top == -1)  
        printf("Empty stack\n");  
    else {  
        int t;
```

```

t = ps->nodes[ps-> top];
ps->top--;
return t;
}
}

```

Una pila vacía no contiene elemento alguno dentro de la estructura y el tope de la misma apunta a nulo.



## Aplicaciones

La estructura pila tienen varias aplicaciones dentro de la ingeniería, de las más conocidas es la que se utiliza dentro de la memoria RAM de un equipo de cómputo. La memoria de las computadoras no es un espacio uniforme, el código que se ejecuta utiliza tres diferentes segmentos de memoria: el texto (text), la pila (stack) y el montículo (heap). Cuando una aplicación inicia, el método principal es invocado y se reserva memoria en la pila o stack. En el segmento de memoria de la pila es donde se alojan las variables requeridas por las funciones del programa. Así mismo, cada vez que se llama una función dentro del programa una sección de la pila, llamada marco o frame, se reserva y es ahí donde las variables de la nueva función son almacenadas. Figura 1. Regiones de la memoria RAM. Cuando una función manda llamar varias funciones, éstas generan un nuevo marco que se va creando uno encima del otro y, cuando las funciones terminan, los marcos se liberan de manera automática en orden inverso (LIFO).

## Referencias

<http://lcp02.fi-b.unam.mx/>

<https://ccc.inaoep.mx/ingreso/programacion/corto2015/Curso-PROPE-PyED-5-Pilas-Colas.pdf>

<http://informatica.uv.es/iiguia/AED/teoria/apuntes/cuatr2/AED.Tema.11.pdf>