

Uso de TADs

Introducción a la Computación (M)

Uso de TADs en Python

TAD Pila

- `CrearPila()` → `Pila(ELEM)`
- `P.apilar(x)`
- `P.vacia()` → `Bool`
- `P.tope()` → `Elem`
- `P.desapilar()`

Python

```
from Pila import *
```

```
p = Pila()  
p.apilar(3)  
print p.tope()  
p.apilar(6)  
print p.tope()
```

```
while not p.vacia():  
    p.desapilar()
```

Ejercicio 1

Usando el TAD Pila implementar en Python una función **reversa** que dada una lista de vuelta sus elementos. Por ejemplo,

`[1,2,3,4] → [4,3,2,1]`

`[] → []`

`[3] → [3]`

Ejercicio 2

Usando el TAD Pila implementar en Python una función que dado un String que contiene una única aparición del carácter '#' determine si es o no capicua

'abc#cba' → True

'abba#abb' → False

'#' → True

Ejercicio

Sea el TAD Heap definido con las operaciones y los órdenes que se muestran a continuación:

TAD Heap

- `Heap()` → `Heap(ELEM)`: Crea un Heap Vacío
- `h.insert(x)`: Agrega un elemento en el Heap en $O(\log(|h|))$
- `h.delMin()` → `Elem`: Devuelve y borra del Heap el elemento más chico en $O(\log(|h|))$

Implementar en Python un algoritmo que ordene los elementos de una lista en `l` en $O(|l| \cdot \log(|l|))$.