

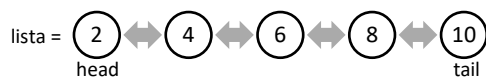
DOCUMENTO TÉCNICO

Implementación de la lista doblemente enlazada

Requisitos

Especificación

Extender la definición de Nodo a tener un apuntador a un nodo previo



Nodo:

```
current = lista.search(6)
```

```
current.get_prev()
```



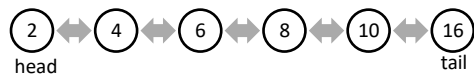
```
current.set_prev(Node(8))
```

```
current.get_prev()
```

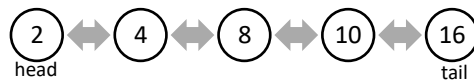


DoubleLinkedList:

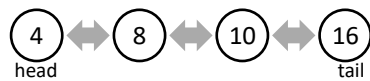
```
lista.insert(16)
```



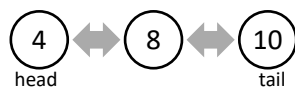
```
lista.delete_element(6)
```



```
lista.delete_element(2)
```



```
lista.delete_element(16)
```



Entrada:

Se reciben los elementos que tendrá la lista enlazada doble separados por espacios, en la siguiente línea se debe ingresar un elemento que este en la lista.

Salida:

El valor del nodo que está antes del elemento dado (función `get_prev()`), el valor al cual ahora el elemento tiene como nodo previo, después de aplicarle la función `set_prev(new_node)`. Y por último, la lista que resulta de aplicarle las funciones `insert(value)` y la función `delete(element)` en los diferentes casos.

Diseño

Estrategia

A los nodos se le agregará el atributo “prev” el cual va a ser un apuntador al nodo previo, junto a este habrán dos nuevas operaciones para leer `get_prev()` y para actualizar `set_prev(new_node)`.

En la lista enlazada doble al insertar un nuevo elemento a este se le asignará como previo el que estaba en la cola de la lista. Y para eliminar un elemento se busca (La función de búsqueda es la misma que la de lista enlazada simple), si no está, no se hace nada y si al contrario si se encuentra, se evalúa dependiendo del caso que presente:

- El elemento por eliminar es el único en la lista, entonces está quedará vacía, head y tail serán None.
- Si el elemento es la cabeza, el que está a continuación de este será la nueva cabeza y tendrá como previo None.
- El elemento es la cola, entonces el valor anterior a este tendrá como siguiente None.
- Por último, si el valor está entre otros dos, el que está seguido al elemento a eliminar tendrá como anterior el nodo previo del elemento.

Adjunto el programa con nombre “DoubleLinkedList”.

Casos prueba

Entrada	Justificación	Salida
2 4 6 8 10 6	Elemento que está entre otros dos	4 9 2 4 6 8 10 16 2 4 6 10 16 4 8 10 16 4 8 10
2 4 6 8 10 2	El elemento por eliminar es la cabeza de la lista	None 9 2 4 6 8 10 16 4 6 10 16 6 8 10 16 6 8 10

Análisis

Temporal

En el mejor de los casos $T(n) = \Omega(1)$

En el peor de los casos $T(n) = O(n)$

Código

Documentación

Dentro del código

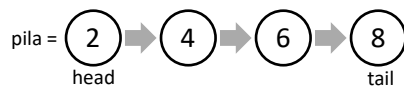
DOCUMENTO TÉCNICO

Implementación de Pila y Cola

Requisitos

Especificación

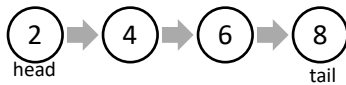
Pila



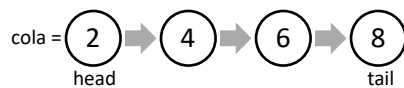
pila.push(10)



pila.pop()



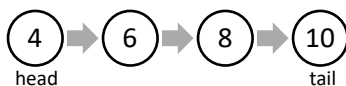
Cola



cola.enqueue(10)



cola.dequeue()



Entrada:

Se reciben los elementos separados por espacios que tendrá la pila la cola, en la siguiente línea el elemento que se insertará.

Salida:

La pila después de insertar el elemento, y de hacer la operación de eliminar (pop).

La cola después de insertar el elemento, y de hacer la operación de eliminar (dequeue).

Diseño

Estrategia

Se va a reutilizar la definición hecha de DoubleLinkedList, en donde tenemos las operaciones de insertar y eliminar.

Para la pila se crea dos funciones: push, la cual insertará un elemento en la cabeza de la pila. Y la función pop que eliminará el elemento que este en la cabeza, de esta manera se sigue la política de LIFO.

En la cola la política es FIFO, por lo tanto, la función de insertar (enqueue) agregará el elemento después del tail, y para eliminar (dequeue) se quita el que este en la cabeza.

Adjunto el programa con nombre "Stack and Queue".

Casos prueba

Entrada	Justificación	Salida
2	Pila y Cola vacías	(Pila) 2 (Cola) 2
2 4	Pila y Cola con un elemento	(Pila) 2 4 (Cola) 2 4
2 4 6	Pila y Cola con dos o más elementos	(Pila) 4 2 (pila.pop()) 2 (pila.push(6)) 6 2 (Cola) 2 → 4 (cola.dequeue()) 4 (cola.enqueue(6)) 4 → 6

Análisis

Temporal

En el mejor de los casos $T(n) = \Omega(1)$

En el peor de los casos $T(n) = O(n)$

Código

Documentación

Dentro del código

Cuarto punto

Adjunto el programa con nombre "4" .