public void InvertirSublista(Lista\_Simple lista1, int primero, int último){

if (primero < 1 || primero >= último || último >= size) {

Best case

return;

}

O(1)

Lista\_Simple aux = new Lista\_Simple();

Nodo it = head;

for (int i = 1; i < primero; i++) {

O(n)

it = it.siguiente;

}

O(1)

Nodo NodoAux = null;

for (int i = primero; i <= último; i++) {

Nodo nuevo = new Nodo(it.clave);

if (aux.head == null) {

aux.head = nuevo;

}

O(n)

else {

NodoAux.siguiente = nuevo;

}

NodoAux = nuevo;

it = it.siguiente;

}

O(1)

aux.Reverse();

it = aux.head;

while(it.siguiente != null){

O(n)

it = it.siguiente;

}

O(1)

Nodo it1 = head;

for (int i = 1; i <= último; i++) {

O(n)

it1 = it1.siguiente;

}

O(1)

it.siguiente = it1;

if (primero == 1){

Best case

head = aux.head;

}

else{

O(1)

it = head;

for (int i = 1; i < primero - 1; i++) {

O(n)

it = it.siguiente;

}

O(1)

it.siguiente = aux.head;

}

}

Varios O(n) + varios O(1)

= Complejidad final: O(n)