Paula Martinez 13866

Algoritmos y Estructura de Datos I

Ejercitación TAD Lista

Ejercicio 1:

linkedlist.py

```python

class LinkedList:
    head=None

class Node:
    value=None
    nextNode=None

def add(L,element):
    #Agrega un elemento al comienzo de la lista
    nodeHead=Node()
    nodeHead.value=element
    current=L.head
    L.head=nodeHead
    if (current==None):
      L.head=nodeHead
    else:
      nodeHead.nextNode=current
      L.head=nodeHead

def search(L,element):
    #Busca un elemento de la lista
    current=L.head
    position=0
    while current!=None:
      if current.value==element:
        return position
      position+=1
      current=current.nextNode
    return None

def insert(L,element,position):
    #Inserta un elemento en una posición determinada de la lista
    cont=0
    current=L.head
    while current!=None:
      cont+=1
      current=current.nextNode
    if (position>cont) or (position<0):
      return None
    if (position==0):
      add(L,element)
      return position
    newNode=Node()
    newNode.value=element
    current=L.head
    for i in range(0,position-1):
      current=current.nextNode
    newNode.nextNode=current.nextNode
    current.nextNode=newNode
    return position

```

```
54 def delete(L,element):
55   #Elimina un elemento de la lista
56   position=search(L,element)
57   current=L.head
58   if (position==None):
59     return None
60   if (position==0):
61     L.head=current.nextNode
62   else:
63     for i in range (0,position-1):
64       current=current.nextNode
65     current.nextNode=current.nextNode.nextNode
66   return position
67
68 def length(L):
69   #Calcula el número de elementos de la lista
70   current=L.head
71   cont=0
72   while current!=None:
73     cont+=1
74     current=current.nextNode
75   return cont
76
77 def access(L,position):
78   #Permite acceder a un elemento de la lista en una posición determinada
79   current=L.head
80   cont=length(L)
81   if (position>=cont) or (position<0):
82     return None
83   for i in range (0,cont):
84     if (i==position):
85       return current.value
86     current=current.nextNode
87
88 def update(L,element,position):
89   #Permite cambiar el valor de un elemento de la lista en una posición determinada
90   cont=length(L)
91   if (position>cont) or (position<0):
92     return None
93   current=L.head
94   for i in range(0,cont):
95     if (position==i):
96       current.value=element
97     current=current.nextNode
98   return position
```

Link Repl.it:
https://replit.com/@Paulonia/Martinez13866U4LinkedList#linkedlist.py

Ejercicio 2:

	Orden de complejidad	
	TAD Lista	**TAD Array**
Add	**O (1)**	**-**
Search	**O (n)**	**O (n)**
Insert	**O (n)**	**O (n)**
Delete	**O (n)**	**O (n)**
Length	**O (n)**	**-**
Access	**O (n)**	**O (1)**
Update	**O (n)**	**-**