

Algoritmos y Estructura de Datos I

Ejercitación TAD PILA (Stack) y COLA (Queue)

Ejercicio 1 (Stack):

```
mystack.py x +
1  from mylinkedlist import *
2
3  def push(S,element):
4      #Agrega un elemento al comienzo de la pila S
5      add(S,element)
6
7  def pop(S):
8      #Extrae el primer elemento de la pila S
9      size=length(S)
10     if (size==0):
11         return
12     element=S.head.value
13     S.head=S.head.nextNode
14     return element
15
```

Ejercicio 2 (Queue):

```
myqueue.py x +
1  from mylinkedlist import *
2
3  def enqueue(Q,element):
4      #Agrega un elemento al comienzo de la cola Q
5      add(Q,element)
6
7  def dequeue(Q):
8      #Extrae el último elemento de la cola Q
9      size=length(Q)
10     if (size==0):
11         return None
12     elif (size==1):
13         element=Q.head.value
14         Q.head=None
15     else:
16         current=Q.head
17         for i in range(0,size):
18             if (i==size-2):
19                 element=current.nextNode.value
20                 current.nextNode=None
21             else:
22                 current=current.nextNode
23     return element
24
```

Ejercicio 3 (Priority Queue):

```
mypriorityqueue.py x +
1  from mylinkedlist import *
2
3  class PriorityQueue:
4      head=None
5
6  class PriorityNode:
7      value=None
8      nextNode=None
9      priority=None
10
11 def enqueue_priority(Q,element,priority):
12     #Agrega un elemento a la cola Q con su prioridad
13     add(Q,element)
14     Q.head.priority=priority
15     return 0
16
17 def dequeue_priority(Q):
18     #extrae el primer elemento de la cola Q con la mayor prioridad
19     size=length(Q)
20     if (size==0):
21         return None
22     maximo=Q.head.priority
23     current=Q.head
24     for i in range(0,size):
25         if (current.priority>=maximo):
26             maximo=current.priority
27             element=current.value
28             position=i
29             current=current.nextNode
30     current=Q.head
31     if (size==1):
32         Q.head=None
33     elif (position==0):
34         Q.head=current.nextNode
35     else:
36         for i in range(0,position-1):
37             current=current.nextNode
38         current.nextNode=current.nextNode.nextNode
39     return element
40
```

Link Repl.it: <https://replit.com/@Paulonia/Martinez13866U4Stack-Queue>