D Y Patil International University

School of Computer Science, Engineering and Applications

Academic Year 2022-2023

Practical Assignment No. 6

Class: F. Y. B. Tech. SEM II **Subject: (CSE 116) Data Structure**

Topic- searching, singly linked List, doubly linked list

Searching

1. Write a python code to implement recursive binary search.

Singly linked list

1. Write a python code to create a single node.

```
class Node:
def __init__(self, data):
self.data = data
self.next = None
n1 = Node(11)
print(n1.data,end=" ")
print(n1.next)
```

2. Write a python code to create an empty a singly linked list.

```
class Node:
  def __init__(self, data):
    self.data = data
    self.next = None
class SinglyLinkedList:
  def __init__(self):
    self.head = None
LL = SinglyLinkedList()
print(LL.head)
```

3. Write a python code to create singly linked list with single node.

Name of the Subject Teacher: Patil Shobhana D.

```
class Node:
  def __init__(self, data):
    self.data = data
    self.next = None
class SinglyLinkedList:
  def __init__(self):
    self.head = None
LL = SinglyLinkedList()
n1 = Node(11)
LL.head = n1
print("A singly linked list is created with single node  with data value")
print(LL.head.data)
```

4. Write a python code to create singly linked list with multiple nodes (3 nodes).

```
class Node:
  def __init__(self, data):
    self.data = data
    self.next = None


class SinglyLinkedList:
  def __init__(self):
    self.head = None
LL = SinglyLinkedList()
n1=Node(11)
LL.head = n1
n2=Node(12)
n1.next=n2
```

Name of the Subject Teacher: Patil Shobhana D.

```
n3=Node(13)

n2.next=n3

print(LL.head.data,end=" ")

print(n2.data,end=" ")

print(n3.data)
```

5. Write a python code to perform traversal operation on singly linked list having 3 nodes

```
class Node:
  def __init__(self, data):
    self.data = data
    self.next = None

class SinglyLinkedList:
  def __init__(self):
    self.head = None
  def traversal(self):
    if self.head is None:
      print("singly list is empty");
    else:
      temp=self.head
      while temp is not None:
        print(temp.data,end=" ")
        temp=temp.next

LL = SinglyLinkedList()
n1=Node(11)
LL.head = n1
n2=Node(12)
n1.next=n2
n3=Node(13)
n2.next=n3
print("Data after Traversal")
LL.traversal()
```

Name of the Subject Teacher: Patil Shobhana D.

6. Write a python code to perform insertion operation on singly linked list having 3 nodes

a. Insertion at beginning

Hint:

```
def insert_at_begining(self,data):
print()
nb=Node(data)
nb.next=self.head
self.head=nb

LL.insert_at_begining(14)
```

b. Insertion at end

Hint:

```
 def insert_at_end(self,data):
    print()
    ne=Node(data)
    temp=self.head
    while temp.next is not None:
      temp=temp.next
    temp.next=ne
```

c. Insertion at a specified position

Hint:

```
def insert_at_specified_node(self,position,data):
    print()
    nib=Node(data)
    temp=self.head
    for i in range(1,position-1):
      temp=temp.next
    nib.next=temp.next
    temp.next=nib

LL.insert_at_specified_node(2,14)
```

7. Write a python code to perform deletion operation on singly linked list having 3 nodes

a. deletion at beginning

Hint:

```
def deletion_at_begining(self):
   print()
   temp=self.head
   self.head=temp.next
   temp.next=None
```

Name of the Subject Teacher: Patil Shobhana D.

b. deletion at end

Hint:

```
def deletion_at_end(self):
 print()
 prev=self.head
temp=self.head.next
while temp.next is not None:
  temp=temp.next
  prev=prev.next
prev.next=None
```

c. deletion at a specified position

Hint:

```
   def deletion_at_specified_node(self,position):
       print()
       prev=self.head
       temp=self.head.next
       for i in range(1,position-1):
         prev=prev.next
       prev.next=temp.next
       temp.next=None

   LL.deletion_at_specified_node(2)
```

8. Write a python code to perform search operation on singly linked list having 3 nodes

Hint:

```
   def search(self,element):
       if self.head is None:
         return False
       curr_node = self.head
       while curr_node!=None:
         if curr_node.data==element:
           return True
         curr_node=curr_node.next
       return False


   if LL.search(11):
     print("Yes")
   else:
     print("No")
```

Name of the Subject Teacher: Patil Shobhana D.

Doubly linked list

1. Write a python code to create a doubly node.
   Hint:

```
class Node:
  def __init__(self, data):
    self.data = data
    self.prev = None
    self.next = None
 n1 = Node(11)
```

2. Write a python code to create an empty a doubly linked list.
   Hint:

```
class Node:
    def __init__(self, data):
    self.data = data
    self.prev = None
    self.next = None
class DoublyLinkedList:
  def __init__(self):
    self.head = None
LL = DoublyLinkedList()
```

3. Write a python code to create doubly linked list with single node.
   Hint:

```
class Node:
    def __init__(self, data):
    self.data = data
    self.prev = None
    self.next = None

class DoublyLinkedList:
  def __init__(self):
    self.head = None

LL = DoublyLinkedList()

n1 = Node(11)
LL.head = n1
```

4. Write a python code to create doubly linked list with multiple node.
   Hint:

Name of the Subject Teacher: Patil Shobhana D.

```
class Node:
  def __init__(self, data):
    self.data = data
    self.prev = None
    self.next = None

class DoublyLinkedList:
  def __init__(self):
    self.head = None
LL = DoublyLinkedList()
n1=Node(11)
LL.head = n1
n2=Node(12)
n1.next=n2
n2.prev=n1
n3=Node(13)
n2.next=n3
n3.prev=n2
```

5. Write a python code to perform traversal operation on doubly linked list having 3 nodes

Hint:

```
  def forward_traversal(self):
    if self.head is None:
      print("Doubly linked list is empty")
    else:
      temp=self.head
      while temp is not None:
        print(temp.data,end=" ")

        temp=temp.next
LL = DoublyLinkedList()
```

6. Write a python code to perform insertion operation on doubly linked list having 3 nodes

a. Insertion at beginning

Hint:

```
  def insertion_at_begining(self,data):
 print()
 nb=Node(data)
temp=self.head
temp.prev=nb
nb.next=temp
 self.head=nb
LL = DoublyLinkedList()
```

Name of the Subject Teacher: Patil Shobhana D.

b. Insertion at end

Hint:

```
def insertion_at_end(self,data):
    print()
    ne=Node(data)
    temp=self.head
    while temp.next is not None:
      temp=temp.next
    temp.next=ne
    ne.prev=temp
```

c. Insertion at specified position

Hint:

```
def insertion_at_specified(self,data,position):
    print()
    nib=Node(data)
    temp=self.head
    for i in range(1,position-1):
      temp=temp.next
    nib.prev=temp
    nib.next=temp.next
    temp.next.prev=nib
    temp.next=nib
```

7. Write a python code to perform deletion operation on doubly linked list having 3 nodes

a. Deletion at beginning

Hint:

```
def deletion_at_begining(self):
    print()
    temp=self.head
    self.head=temp.next
    temp.next=None
    self.head.prev=None
```

b. Deletion at end

Hint:

```
def deletion_at_end(self):
    print()
    temp=self.head.next
    before=self.head
```

Name of the Subject Teacher: Patil Shobhana D.

```
        while temp.next is not None:
          temp=temp.next
          before=before.next
        before.next=None
        temp.prev=None
```

c. Deletion at specified position

Hint:

```
def deletion_at_specified_position(self,position):
    print();
    temp=self.head.next
    before=self.head
    for i in range(1,position-1):
        temp=temp.next
        before=before.next
    before.next=temp.next
    temp.next.prev=before
    temp.next=None
    temp.prev=None
```

8. Write a python code to perform search operation on doubly linked list having 3 nodes

Hint:

```
def search_node(self, val_to_search):
    i = 1;
    flag_val = False;
    curr = self.head;
    if(self.head == None):
        print("List is empty")
        return
    while(curr != None):
        if(curr.data == val_to_search):
            flag_val = True
            break
        curr = curr.next
        i = i + 1
    if(flag_val):
        print("The node is present in the list at position : ")
        print(i)
    else:
        print("The node isn't present in the list")
```

Name of the Subject Teacher: Patil Shobhana D.