Doubly Linked List
Metode: ADDnodeLast(node)

Før: A ↔ B ↔ C
Efter: A ↔ B ↔ C ↔ D

Kode:
```
old = tail
tail = node
tail.previous = old
old.next = tail
```

ADD Node First (node)
Før: A ↔ B ↔ C
Efter: D ↔ A ↔ B ↔ C

Kode:
```
old = head
head = node
head.next = old
old.prev = head
```

ADD last + Add First: (data)
Samme som ovenfor, med denne linje
først:
```
node = new Node(data)
```

Tilføjelse:
```
if (old){...}
else: head/tail = tail/head
```

Remove last:

Før: A — B — C

Efter: A — B

```
          if tail.prev:
Kode:     tail.prev.next = null
          tail = tail.prev

          else: tail = null
                head = null
```

Remove first:

Samme som ovenfor, bare omvendt

```
Remove Node (node)
    current = head
    while current:
        if current==node:
            if current == head
                remove first()
            elif current == tail:
                remove Last()
            else:
                current.next.prev = cur
                current.prev.next = current
    current = current.next
```

Insert before node (New, existing)

For: A - B - C
Efter: A - B - D - C

```
Kode: if existing == head:
          add node first (new)

      else:
          New.prev = existing.prev
          New.next = existing
          existing.prev.next = new
          existing.prev = new
```

Insert after node (New, existing)

```
Kode:  if existing == tail:
           add node last (New)
       else:
           New.prev = existing
           New.next = existing.next
           existing.next.prev = New
           existing.next = New
```

```
  Swap Nodes (a, b)              Rettelse:
Før:   A - B - C - D             OldA = A.data
Efter: B - A - C - D             A.data = B.data
                                 B.data = oldA
Kode: old prev A = A.prev
      old Next A = A.Next
      if A.Next == B:
         A.Prev = B
         A.Next = B.Next          B.Next.Prev = A
         B.Next = A
         B.Prev = old Prev A      old Prev A.next =
     elif B.Next == A
         A.Next = B
         A.Prev = B.Prev          B.Prev.Next = A
         B.Next = old Next A      old Next A.Prev = B
         B.Prev = A
    else:
         A.Next = B.Next          B.Next.Prev = A
         A.Prev = B.Prev          B.prev.Next = A
         B.Next = old Next A      ONA.Prev = B
         B.Prev = old Prev A      OPA.Next = B
    if A == head:
        head = B
    elif B == head:
        head = A
    if A == tail:
       tail = B
    elif B == tail:
       tail = A
```