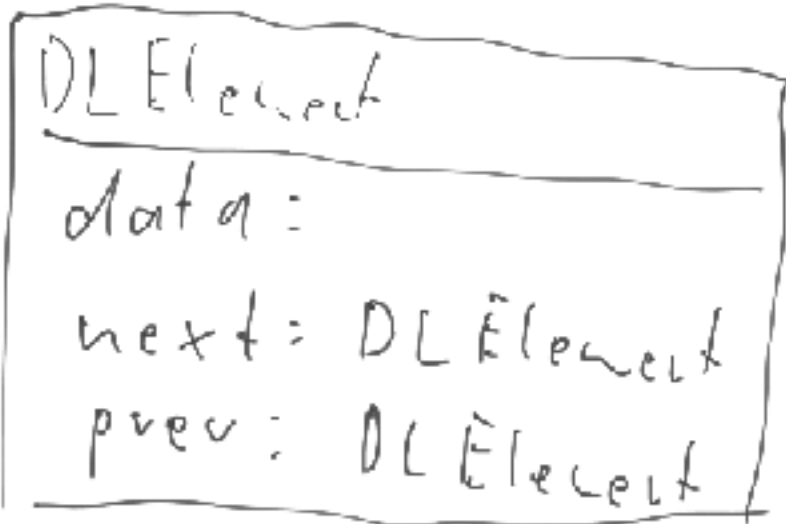
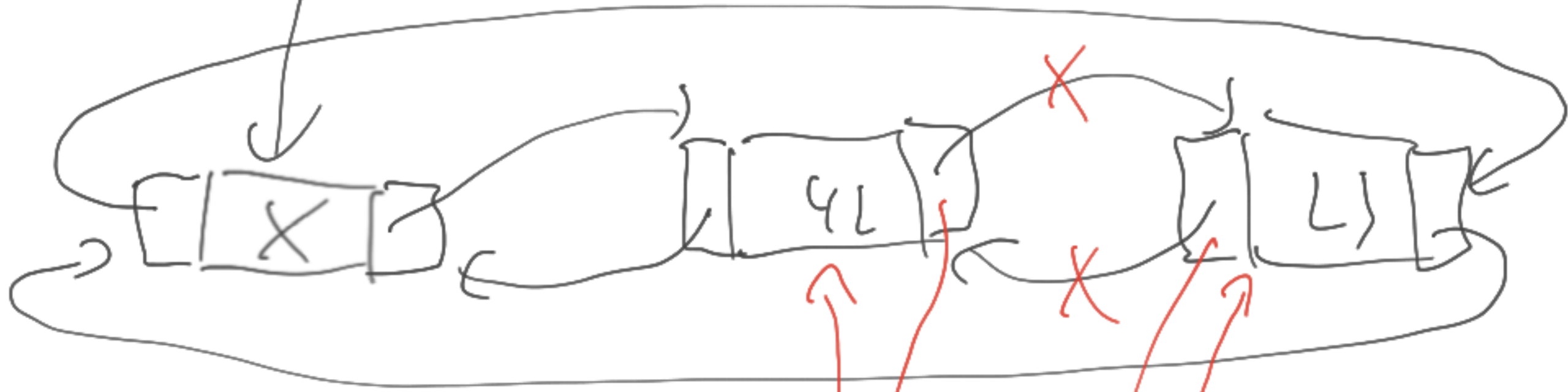
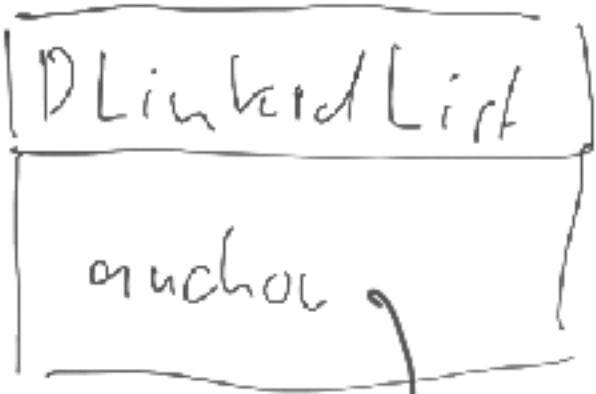
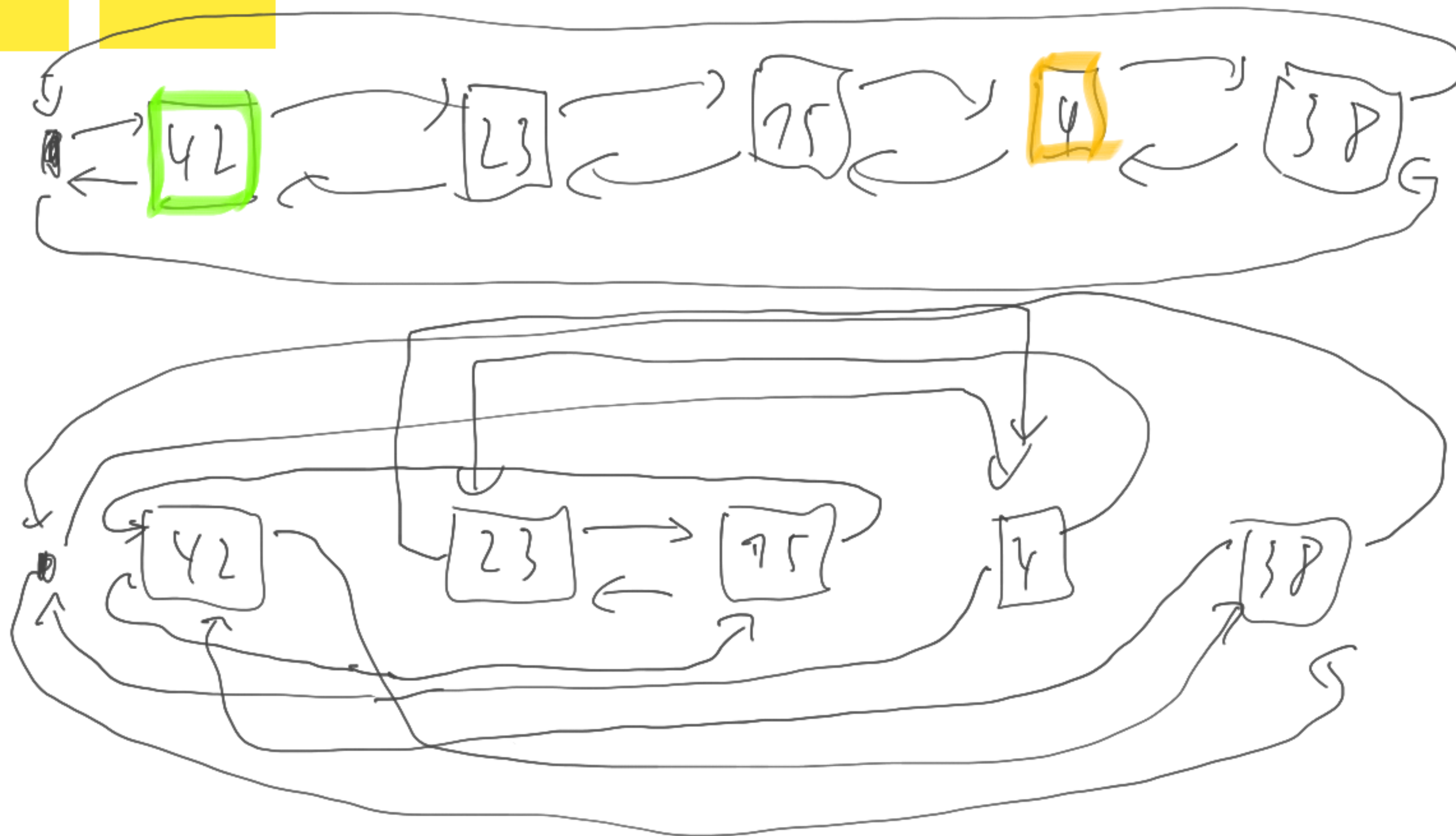


Doppelt
verkettete
Liste



Vertauschen
von
Elementen

"swap"



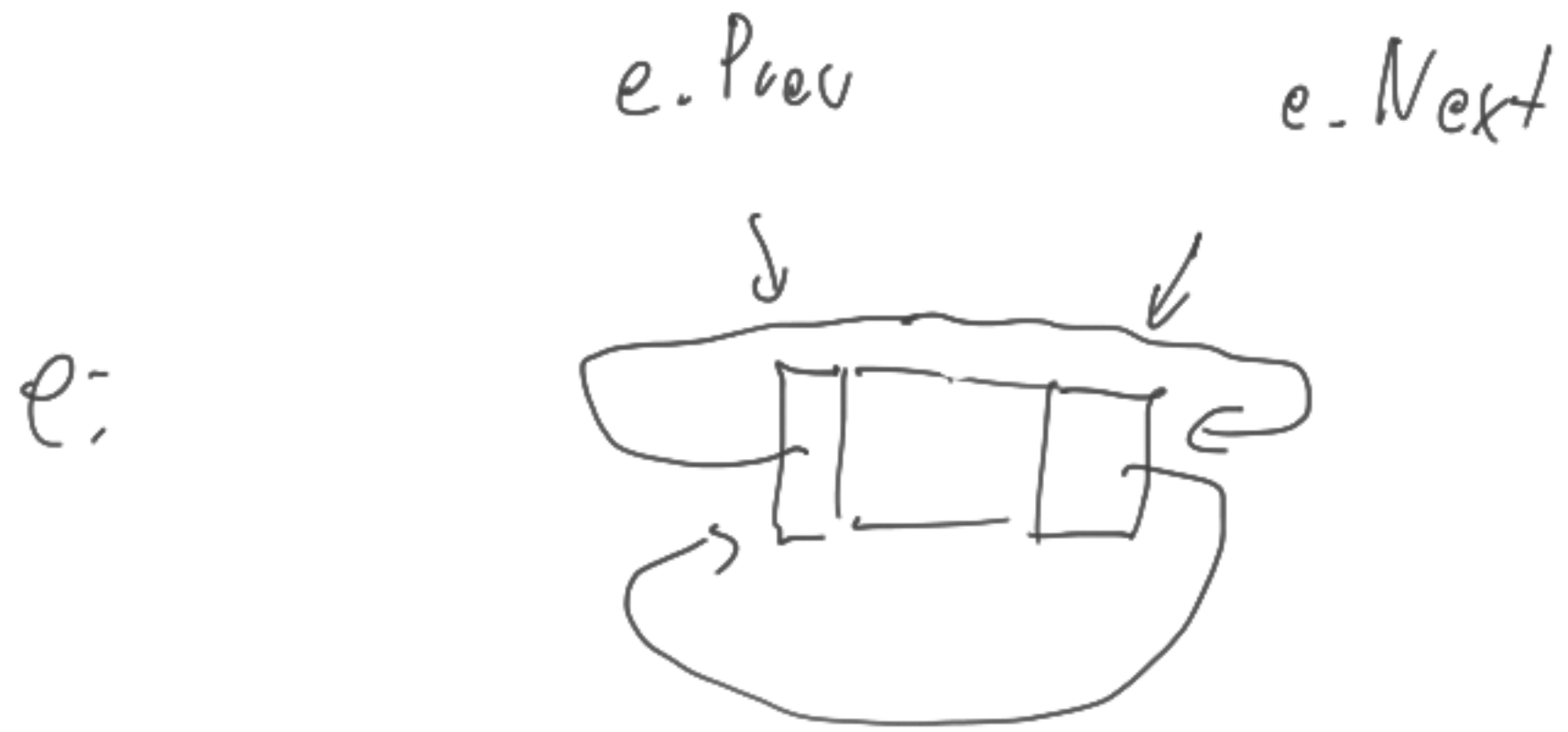
**Rekursive
Definition
Liste**

**Eine
Liste
ist ...**

**...
leer**

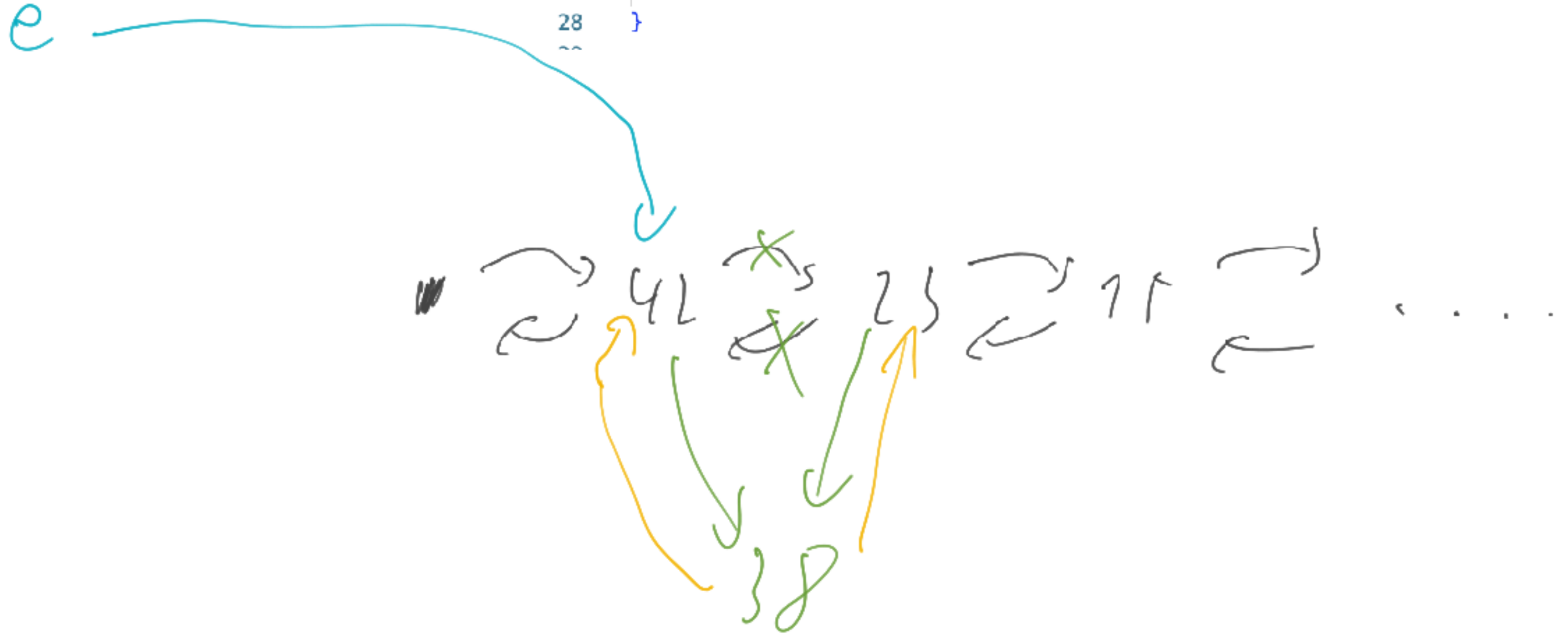
**... ein
Element
plus eine
Liste**

**Head
und
Tail**



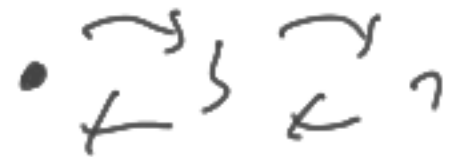
Insert

```
22 // Insert inserts a new element after the current element.  
23 func (e *DLElement) Insert(value int) *DLElement {  
24     newElement := &DLElement{Prev: e, Value: value, Next: e.Next}  
25     e.Next.Prev = newElement  
26     e.Next = newElement  
27     return newElement  
28 }
```



Fehlerhafter Test für Insert

e



```
20 func TestDLElement_Insert(t *testing.T) {  
21     e := NewDLElement()  
22     e.Insert(1)  
23     e.Insert(3)  
24     e.Insert(2)  
25  
26     if e.Next.Value != 1 {  
27         t.Error("e.Next.Value should be 1, but is", e.Next.Value)  
28     }  
29 }
```

e.Insert(1)

e.Insert(3)

e.Insert(2)