

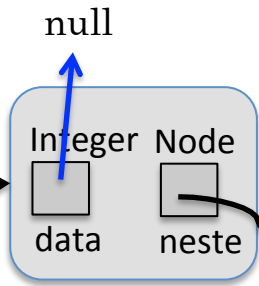
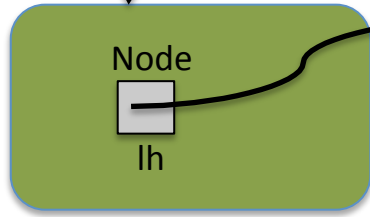
# Quicksort

## Et eksempel på rekursjon

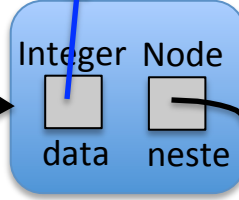
INF1010

Stein Michael Storleer

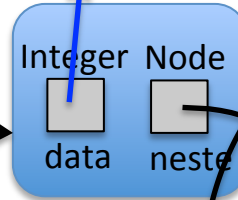
Lenkeliste<Integer>



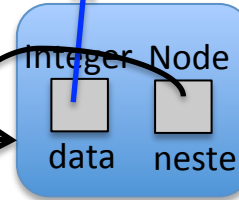
5



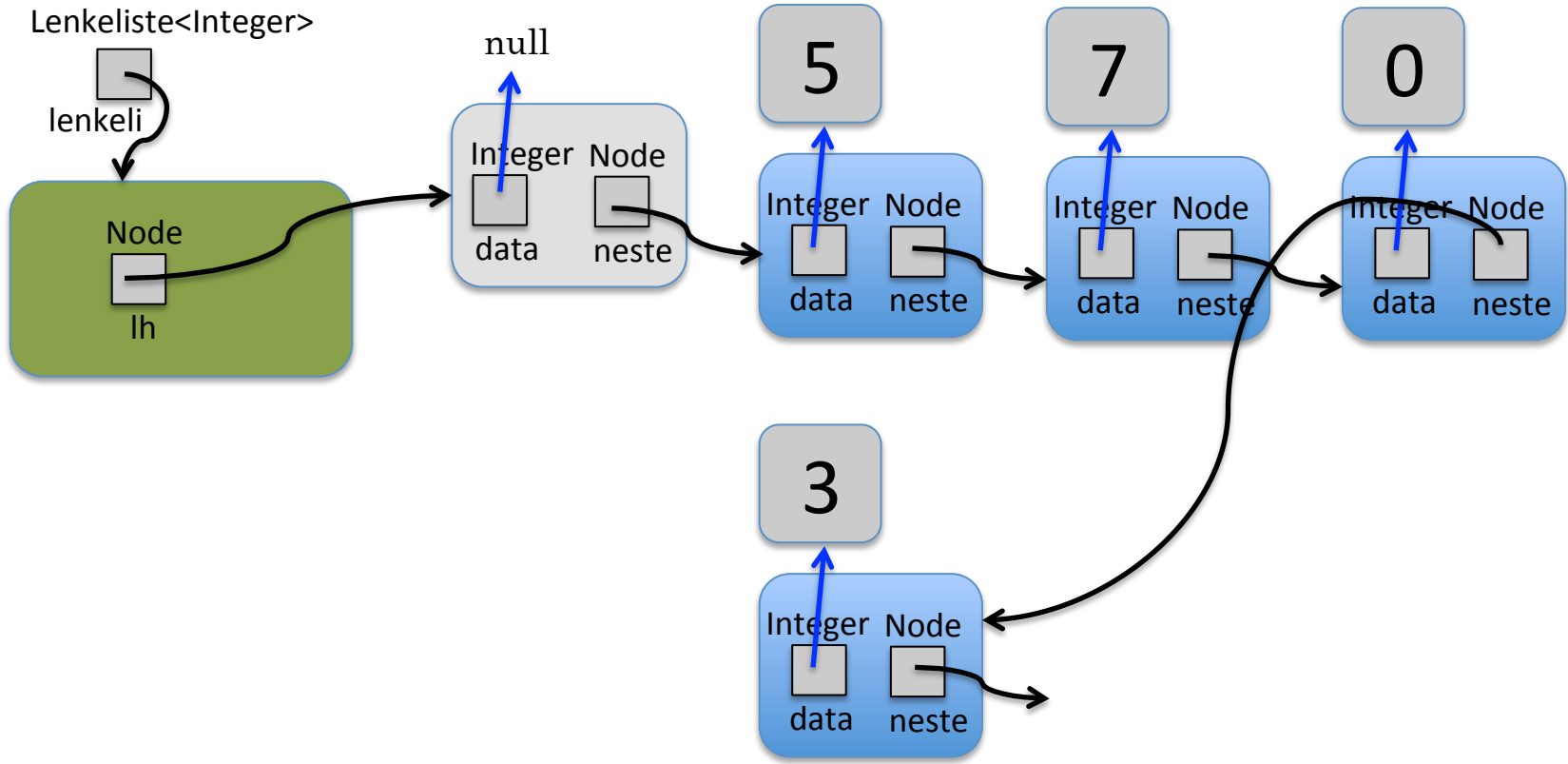
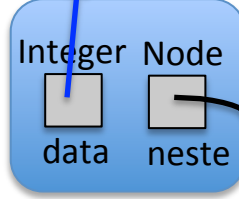
7



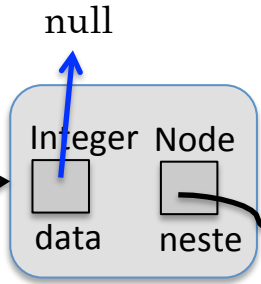
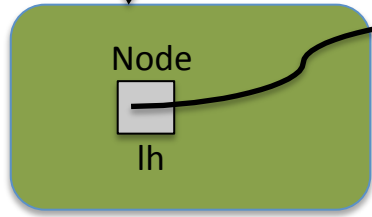
0



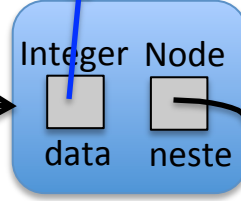
3



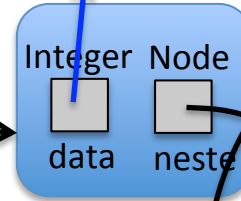
Lenkeliste<Integer>



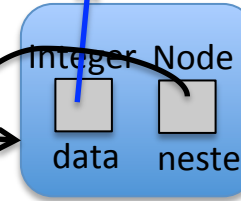
5



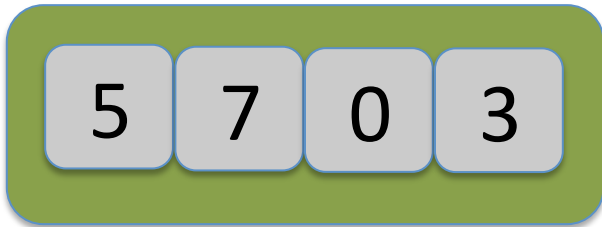
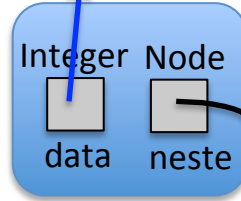
7



0



3



5

7

0

3

9

9

1

8

4

2

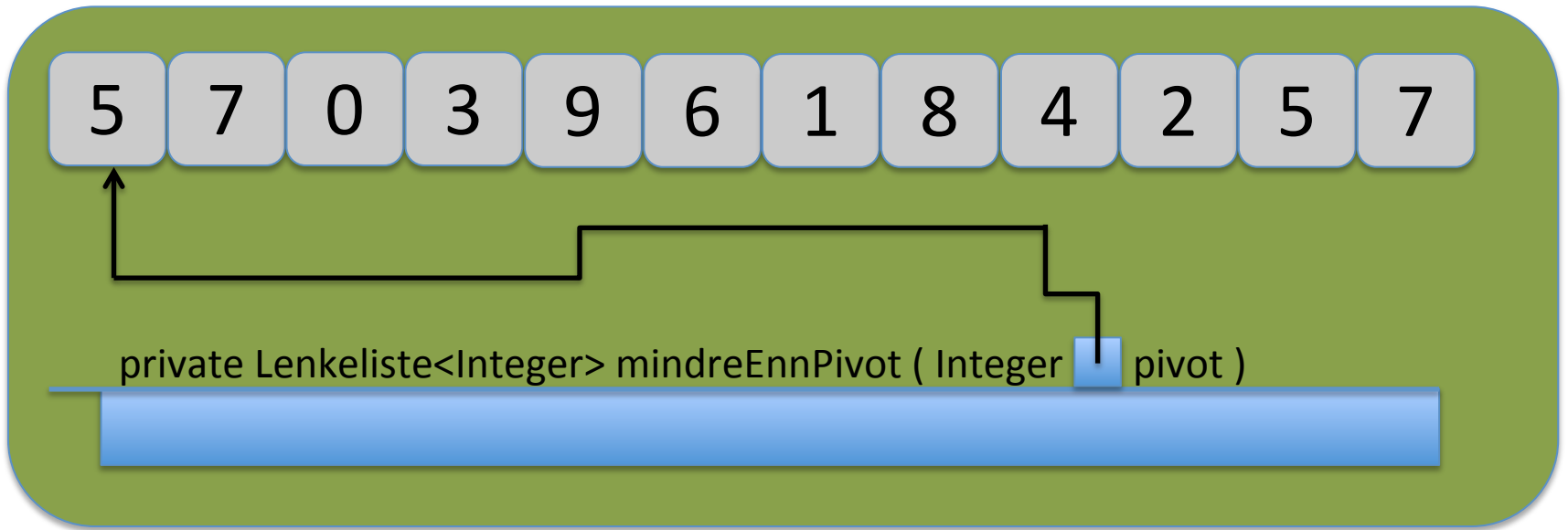
7

7

5 7 0 3 9 6 1 8 4 2 5 7

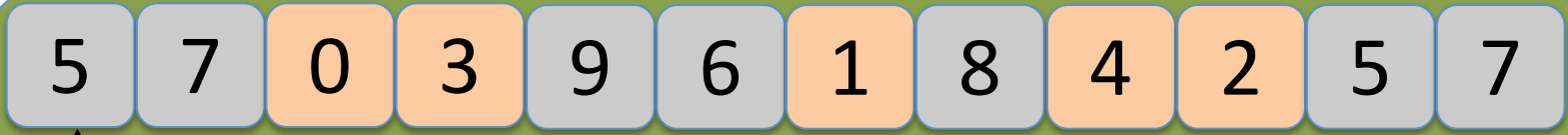
```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```

```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```



```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```



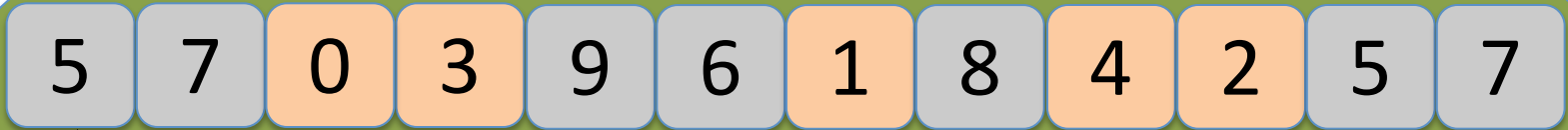


```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```

Lenkeliste<Integer>



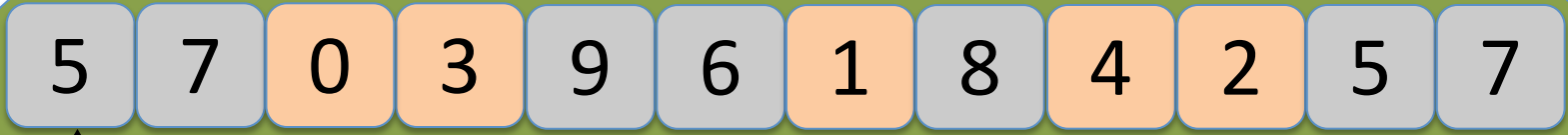




```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```

Lenkeliste<Integer>

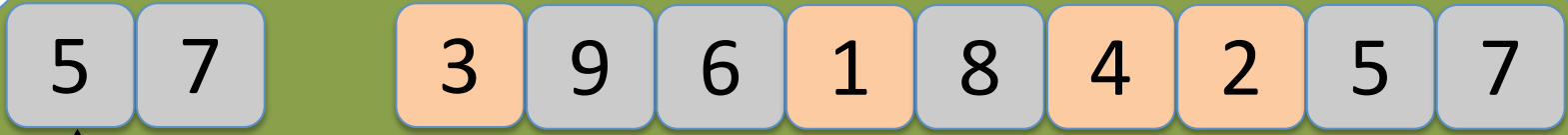




```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```

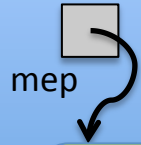
Lenkeliste<Integer>





```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```

Lenkeliste<Integer>



0



```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```

Lenkeliste<Integer>



5 7

9 6

8 4 2 5 7

```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```

Lenkeliste<Integer>



1 3 0

5 7

9 6

8

2 5 7



```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```

Lenkeliste<Integer>



4 1 3 0

5 7

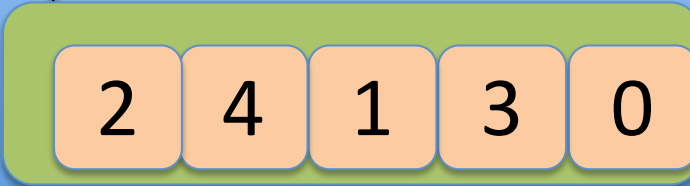
9 6

8

5 7

```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```

Lenkeliste<Integer>



5 7

9 6

8

5 7

```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```

Lenkeliste<Integer>



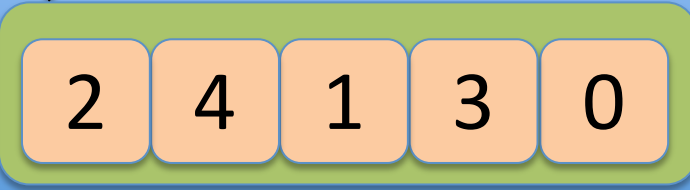
2 4 1 3 0





```
private Lenkeliste<Integer> mindreEnnPivot ( Integer pivot )
```

Lenkeliste<Integer>



5

7

0

3

9

6

1

8

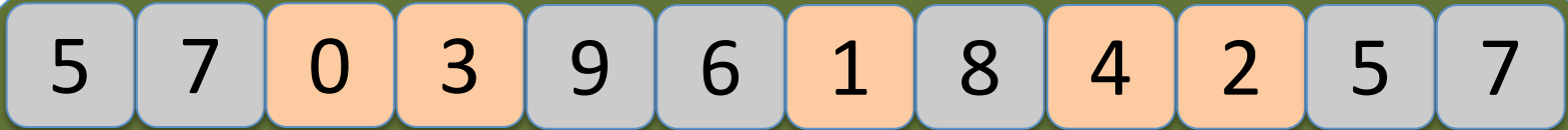
4

2

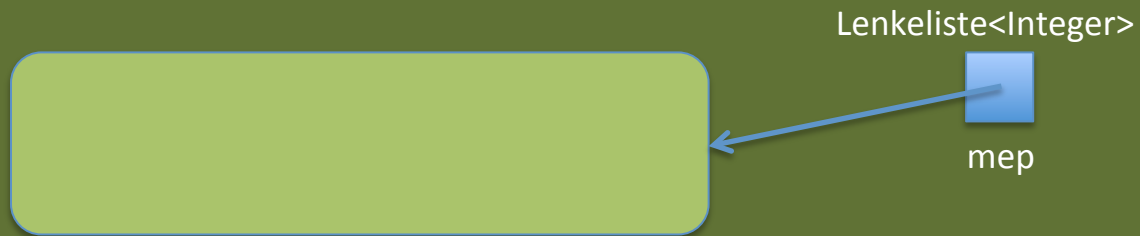
5

7

```
private Lenkeliste<T> mindreEnnPivot(T pivot) {  
    Lenkeliste<T> mep = new Lenkeliste<T>();  
    Iterator<T> it = iterator();  
    while (it.hasNext()) {  
        T t = it.next();  
        if ( t.compareTo(pivot) <= 0 ) {  
            mep.settInnForan(t);  
            it.remove();  
        }  
    }  
    return mep;  
}
```

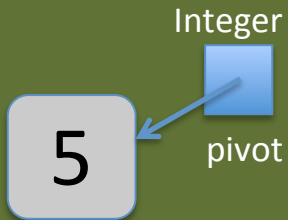


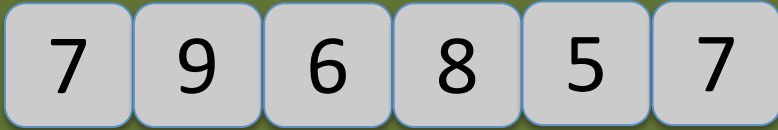
```
Lenkliste<T> mep = new Lenkliste<T>();
```



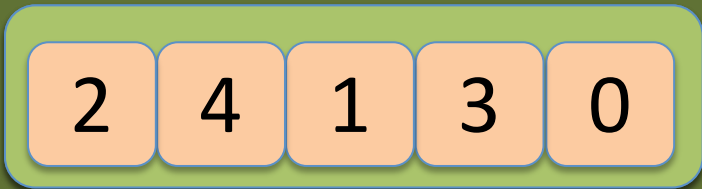
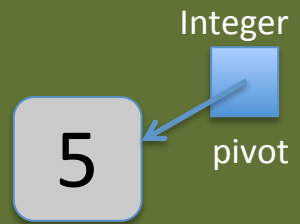


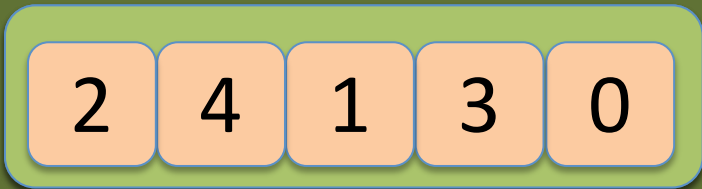
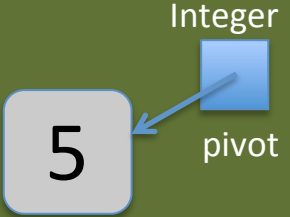
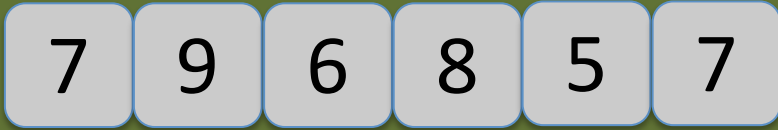
```
Lenkeliste<T> mep = new Lenkeliste<T>();  
T pivot = this.taUtForan();
```

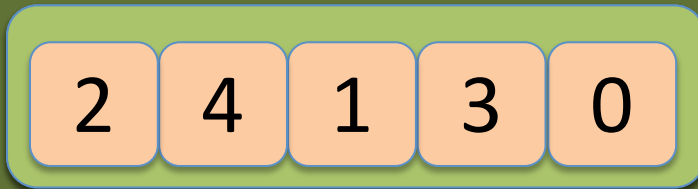




```
Lenkeliste<T> mep = new Lenkeliste<T>();  
T pivot = this.taUtForan();  
mep = mindreEnnPivot(pivot);
```







Lenkeliste<Integer>



mep





Lenkeliste<Integer>



mep





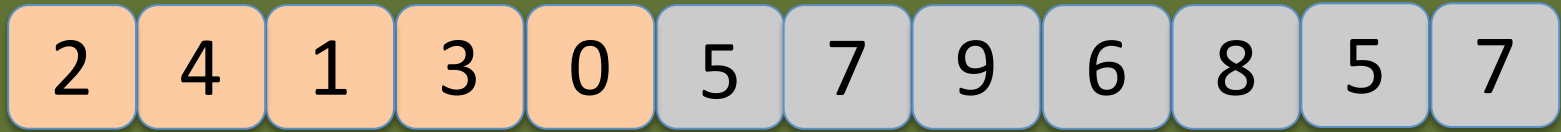


Lenkeliste<Integer>



mep

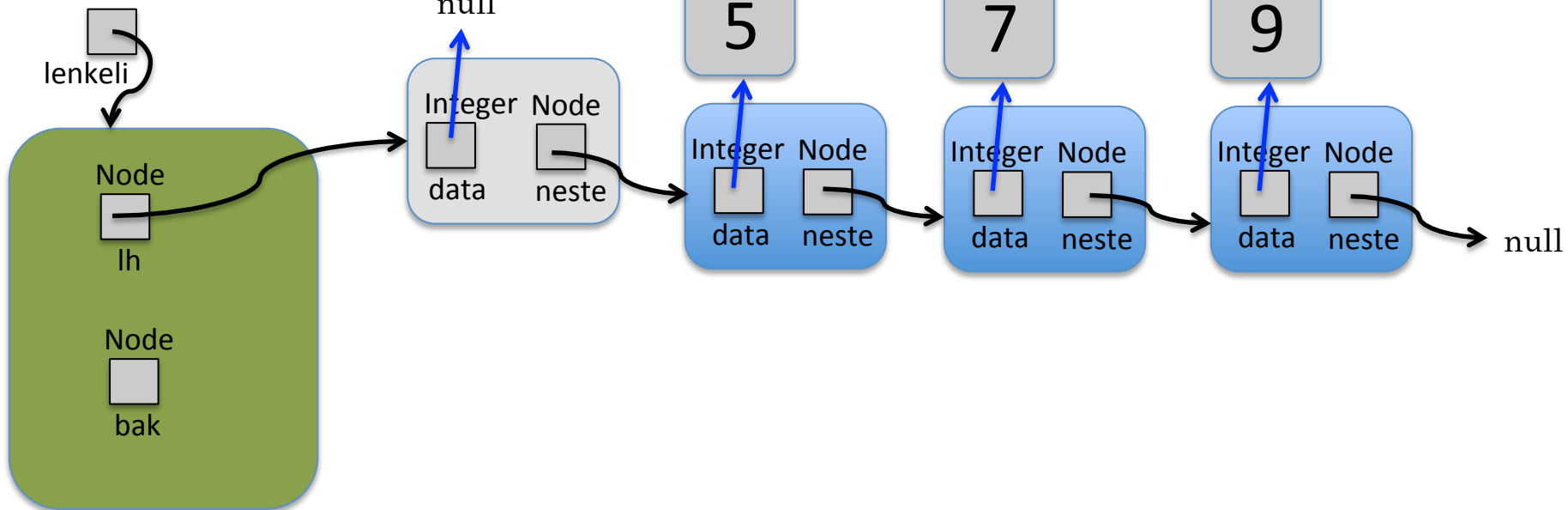




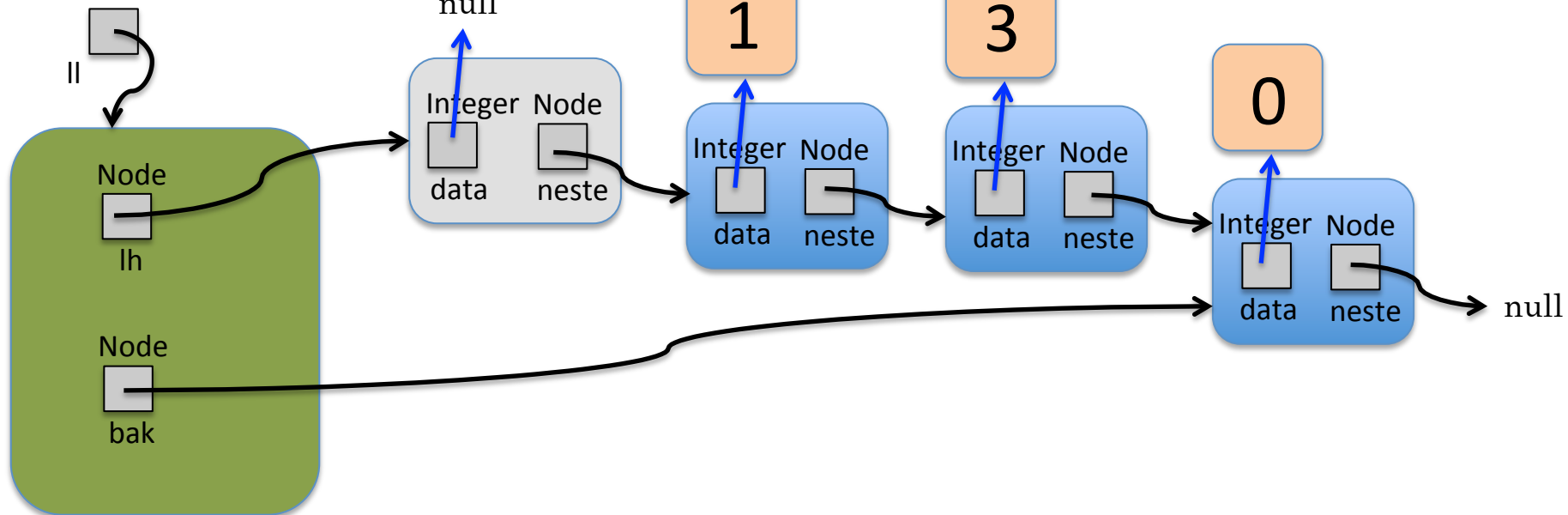
små verdier

store verdier

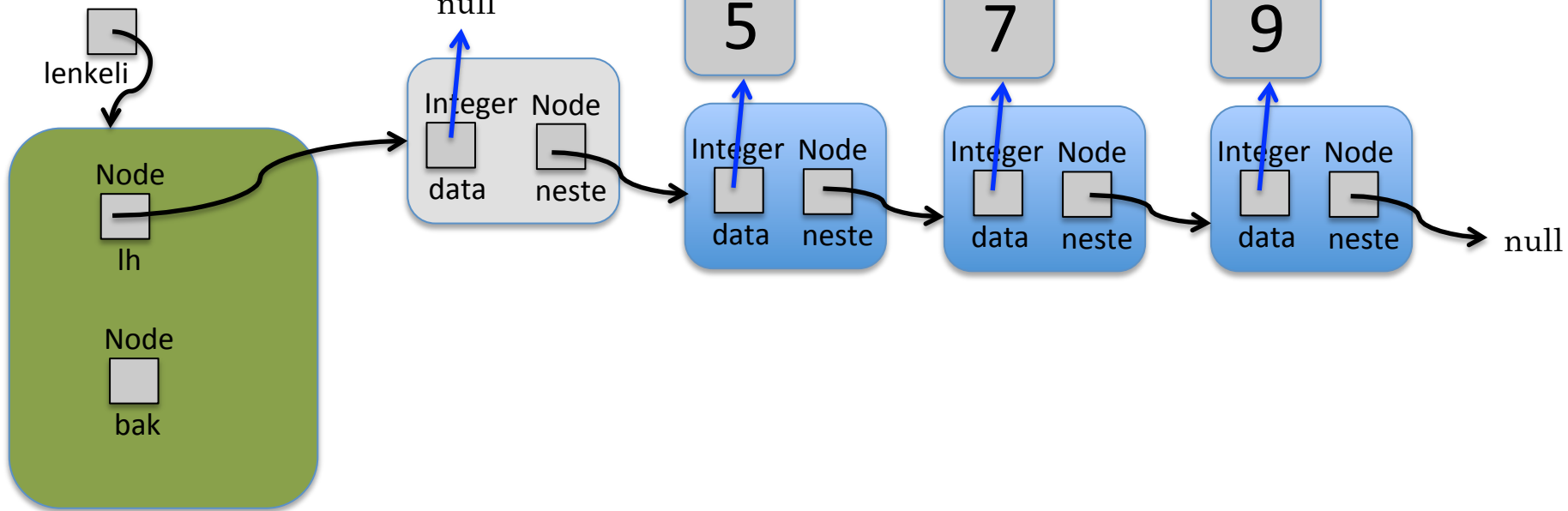
Lenkeliste<Integer>



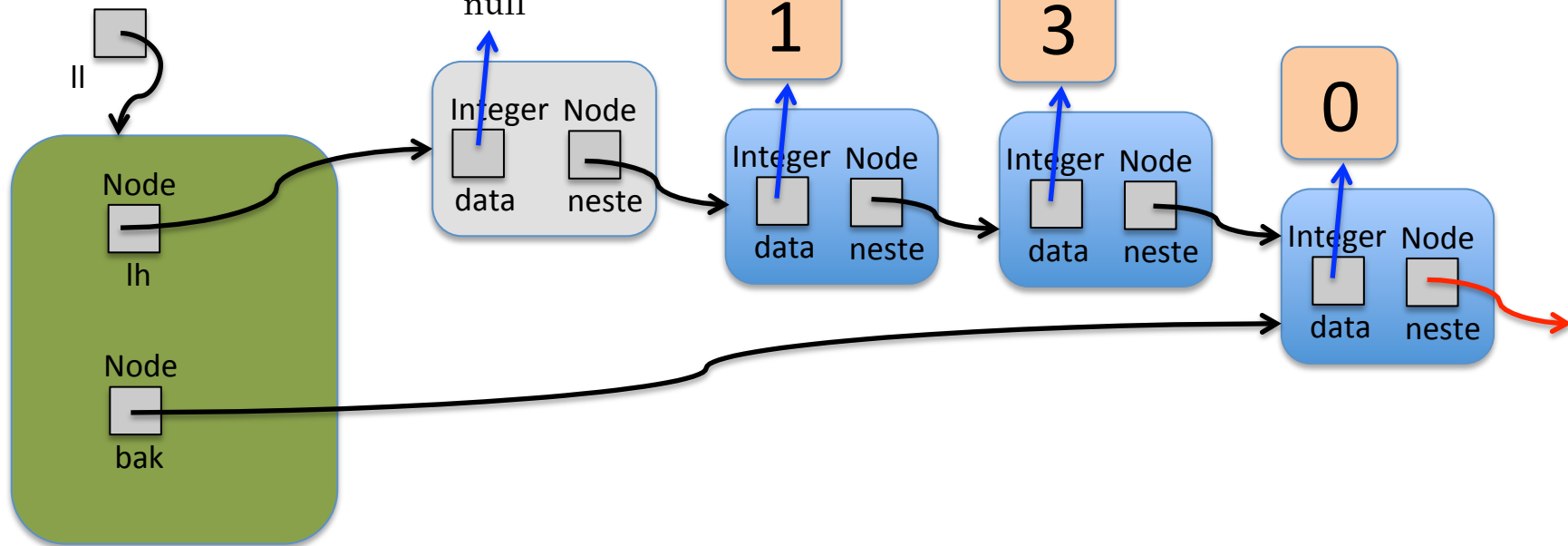
Lenkeliste<Integer>



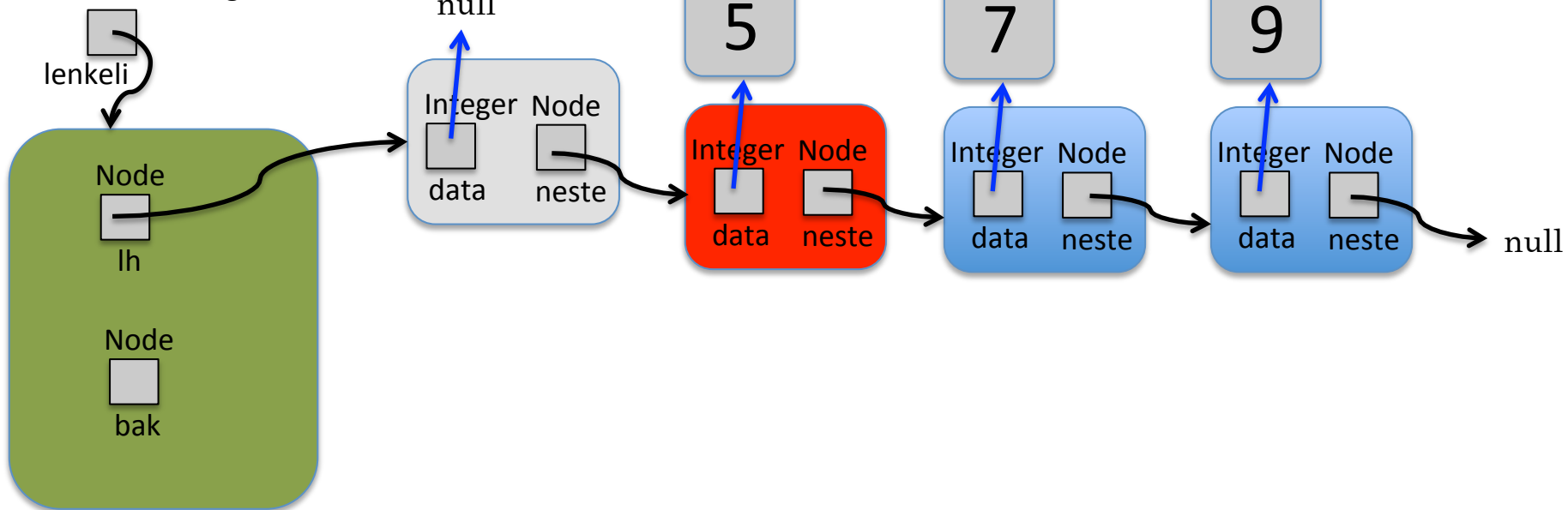
Lenkeliste<Integer>



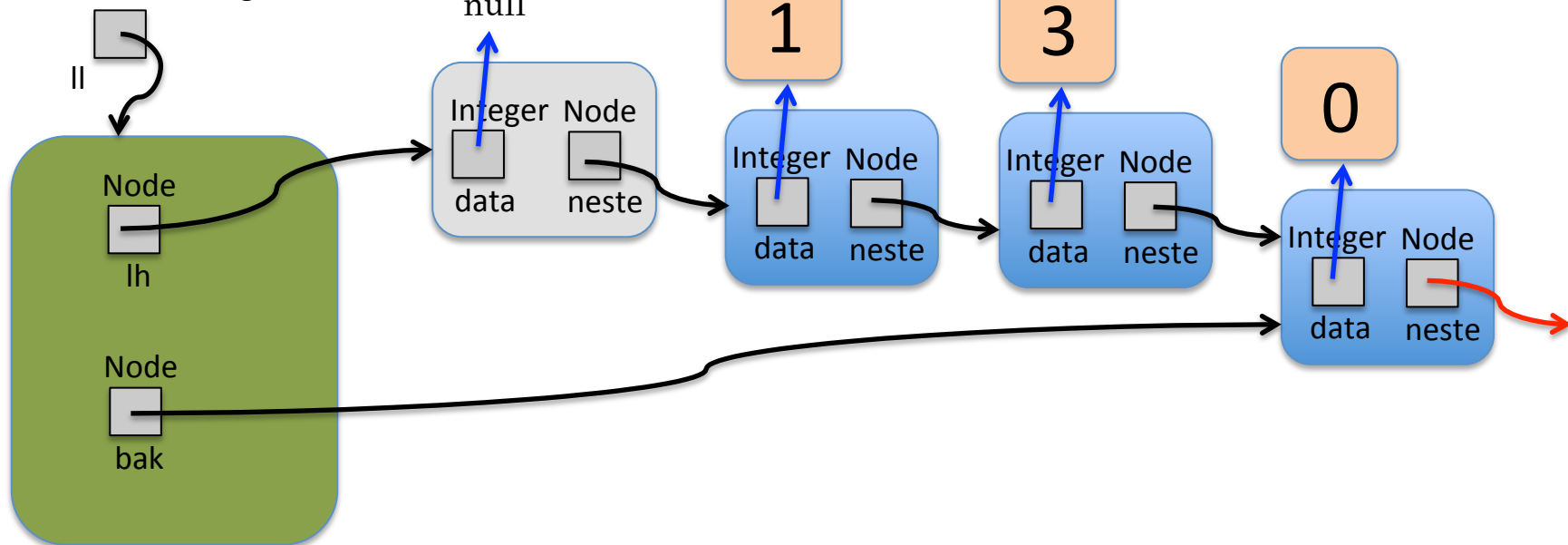
Lenkeliste<Integer>



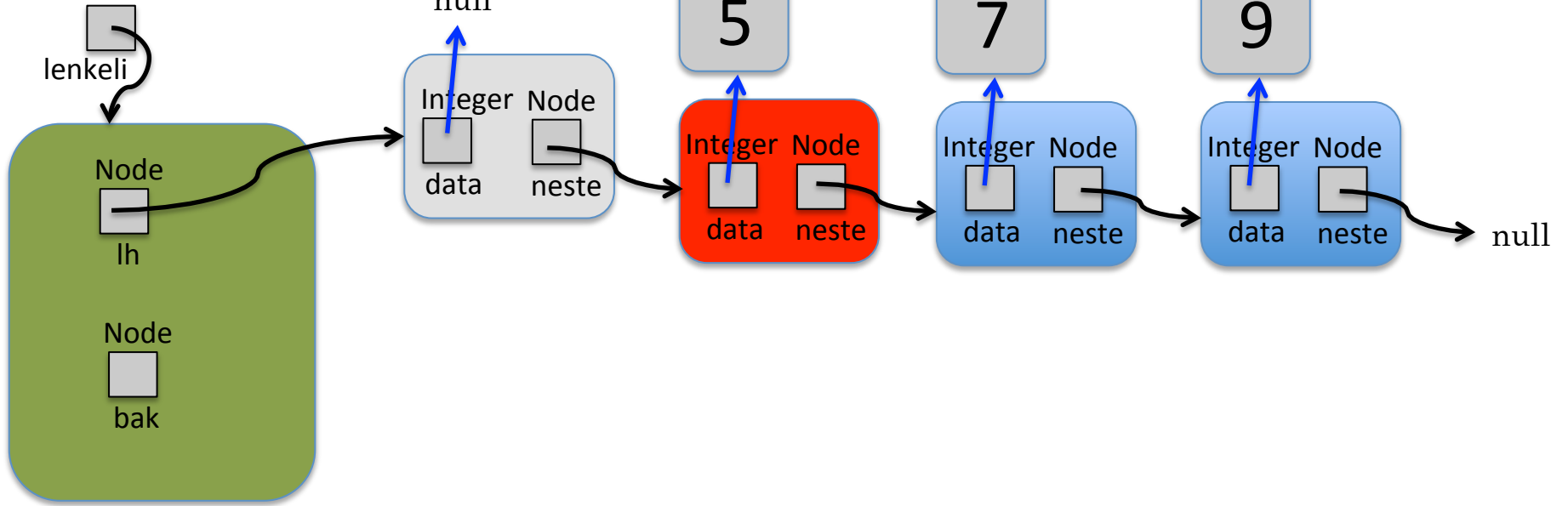
Lenkeliste<Integer>



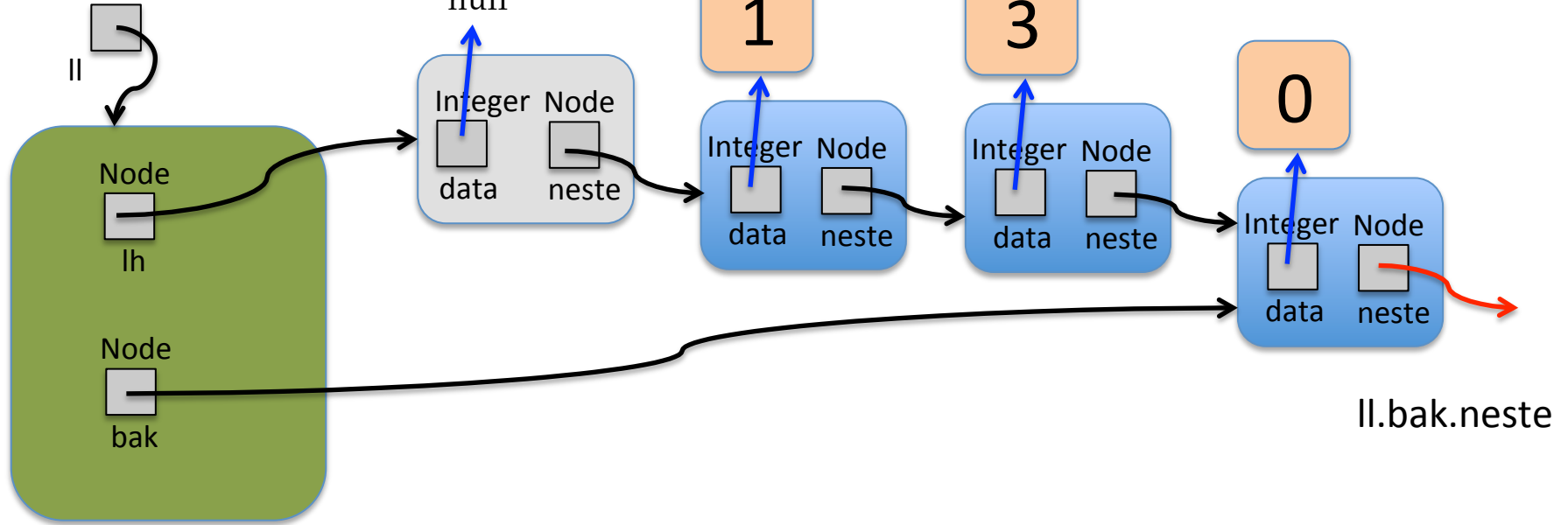
Lenkeliste<Integer>



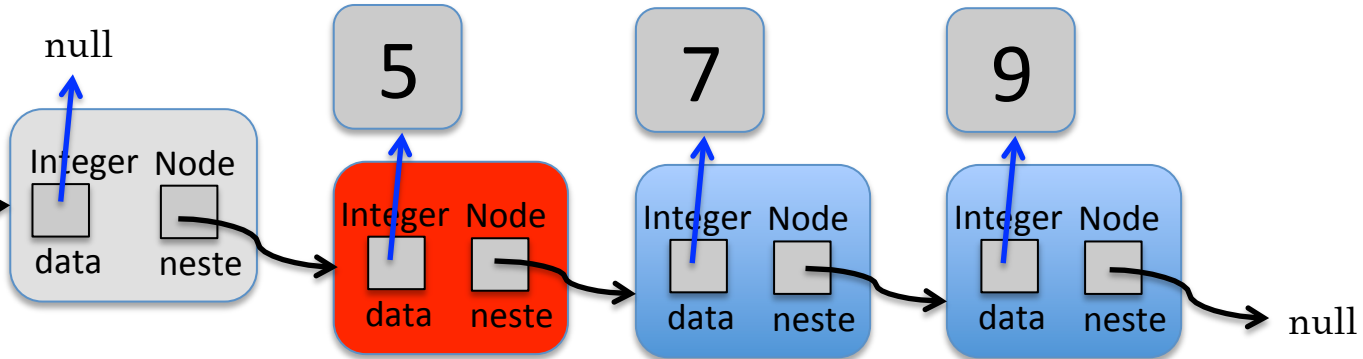
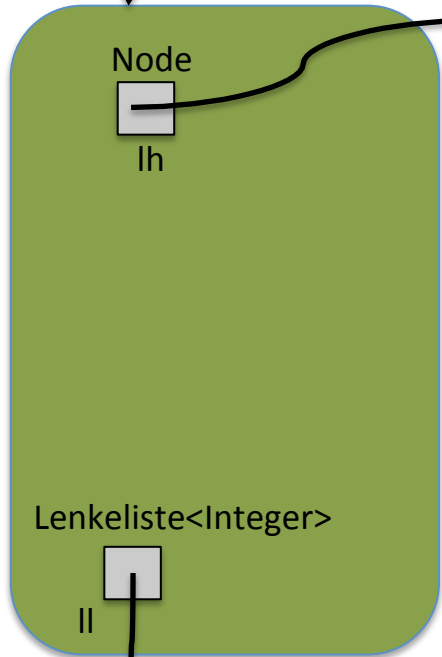
Lenkeliste<Integer>



Lenkeliste<Integer>

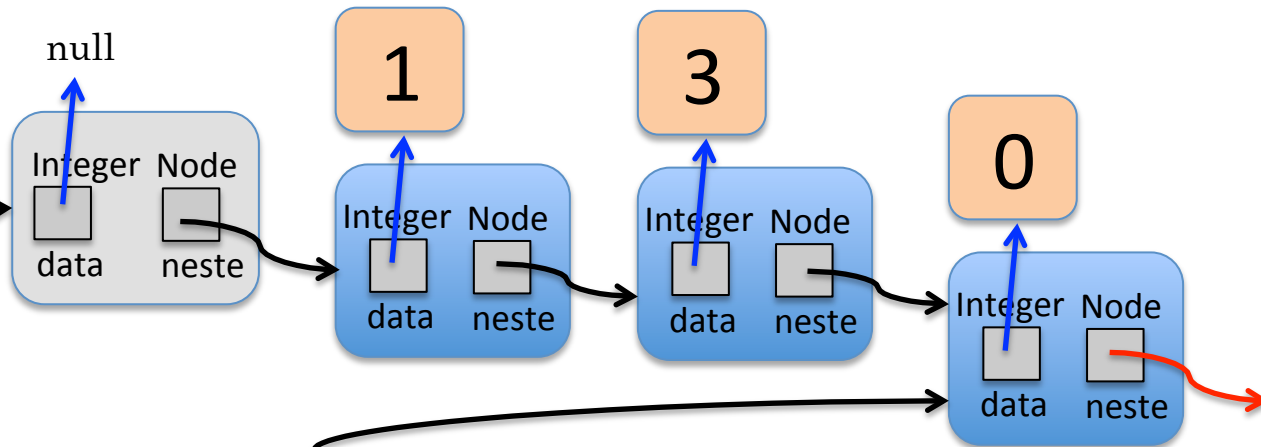
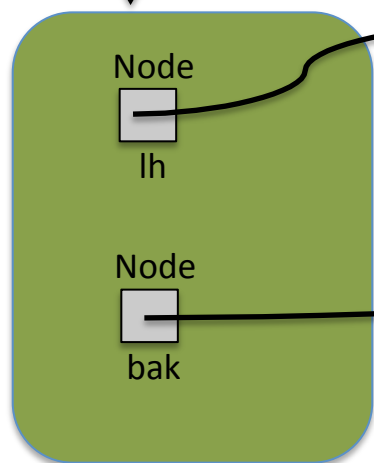


Lenkeliste<Integer>



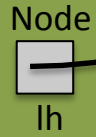
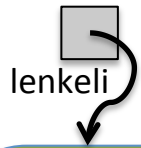
lenkeli.lh.neste

Lenkeliste<Integer>



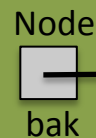
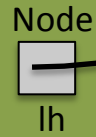
ll.bak.neste

Lenkeliste<Integer>

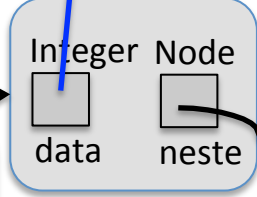


ll.bak.neste = lh.neste;

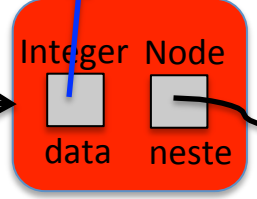
Lenkeliste<Integer>



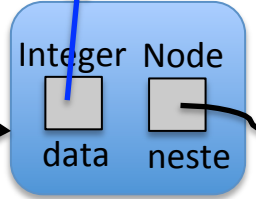
null



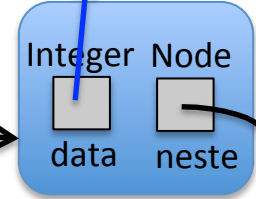
5



7

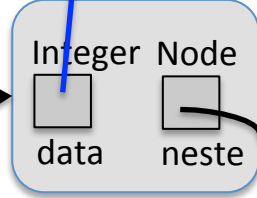


9

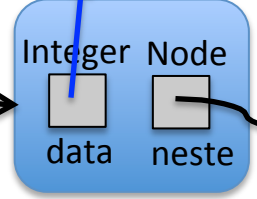


null

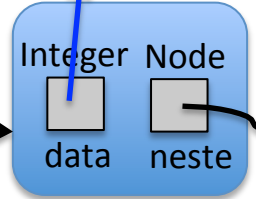
null



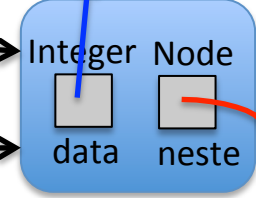
1



3

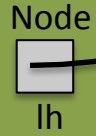
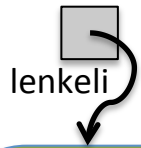


0



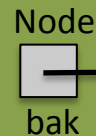
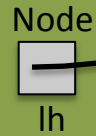


Lenkeliiste<Integer>

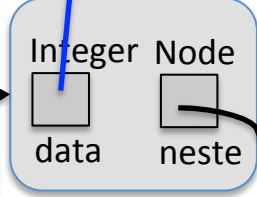


ll.bak.neste = lh.neste;

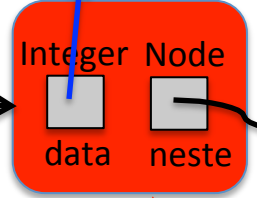
Lenkeliiste<Integer>



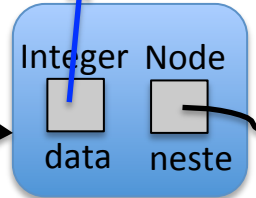
null



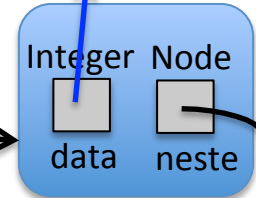
5



7

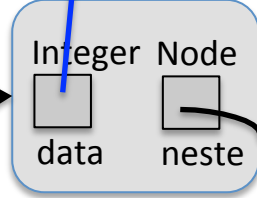


9

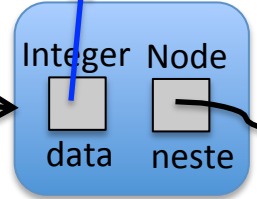


null

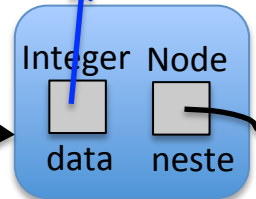
null



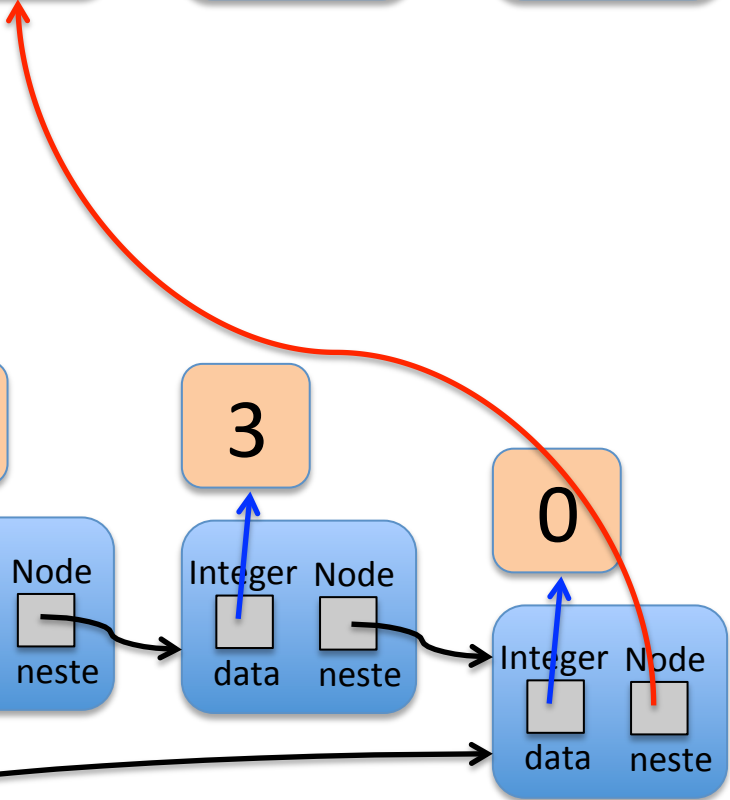
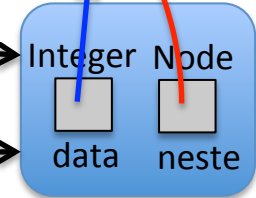
1



3



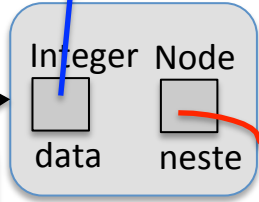
0



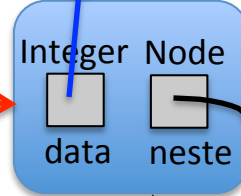
Lenkeliste<Integer>



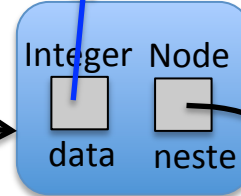
null



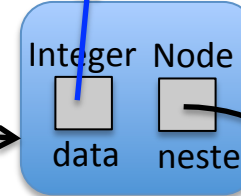
5



7



9



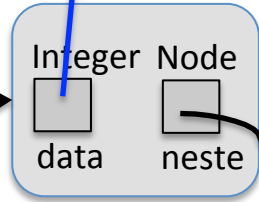
null

Node  
lh

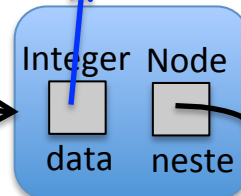
ll.bak.neste = lh.neste;

Lenkeliste<Integer>  
ll

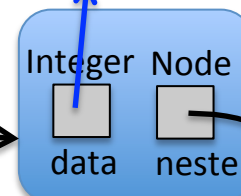
null



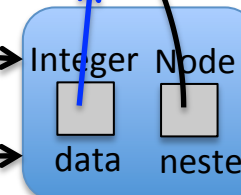
1



3

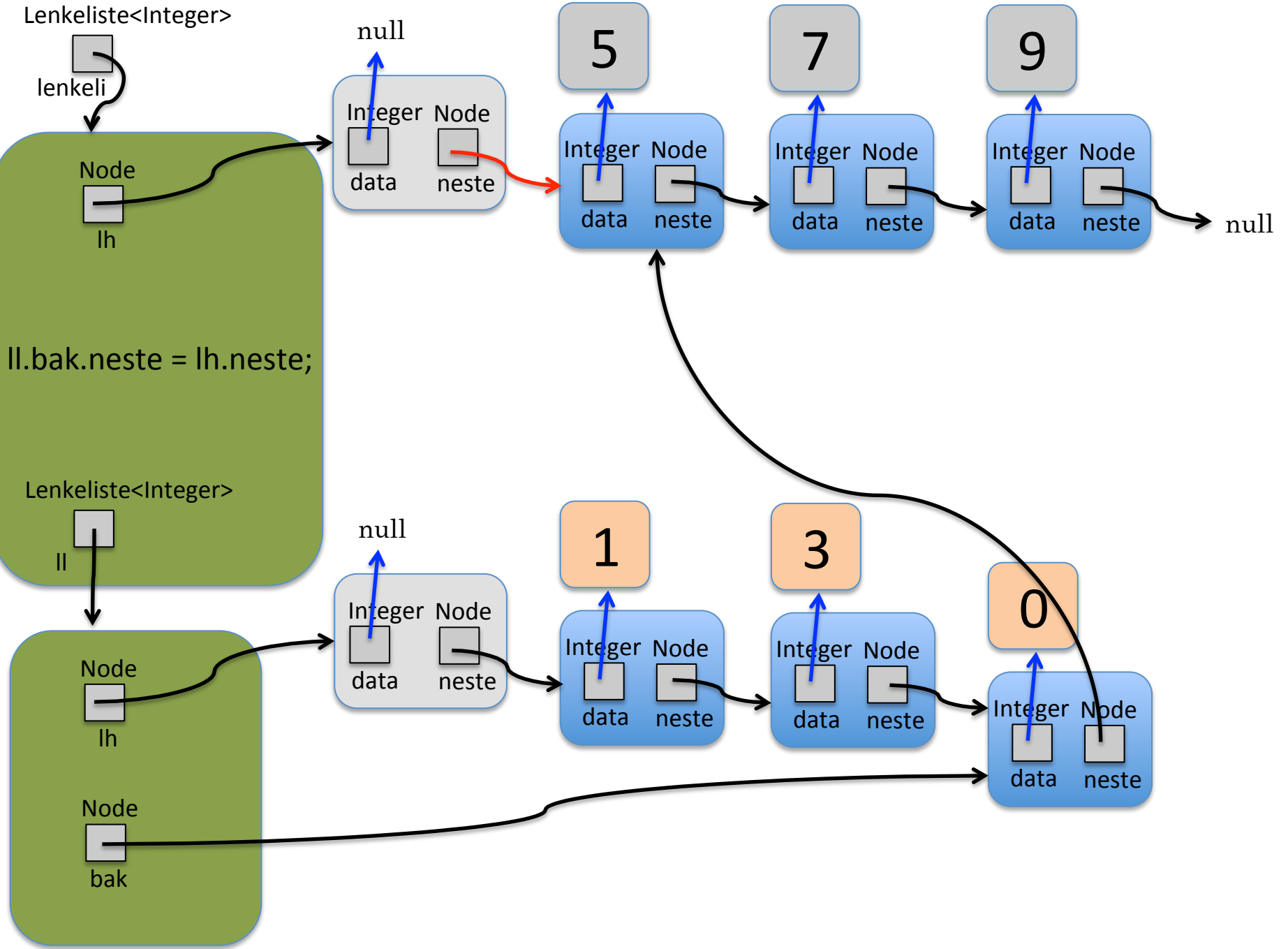


0

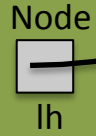
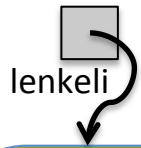


Node  
lh

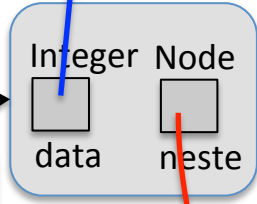
Node  
bak



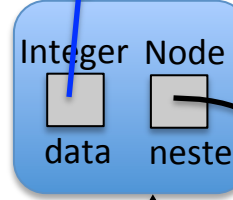
Lenkeliste<Integer>



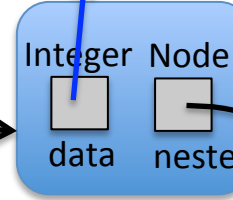
null



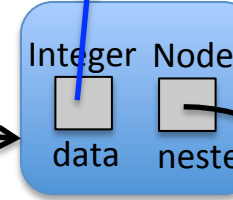
5



7



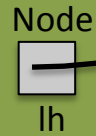
9



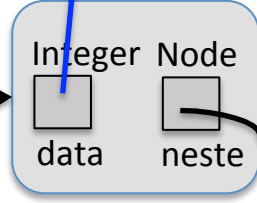
null

ll.bak.neste = lh.neste;  
lh.neste = ll.lh.neste;

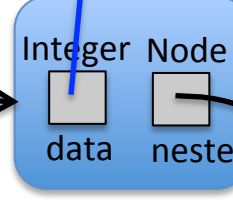
Lenkeliste<Integer>



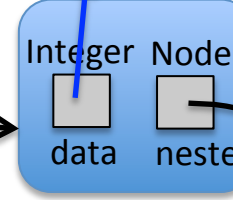
null



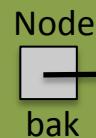
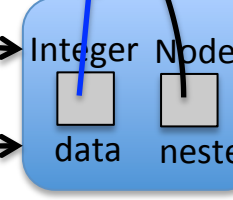
1



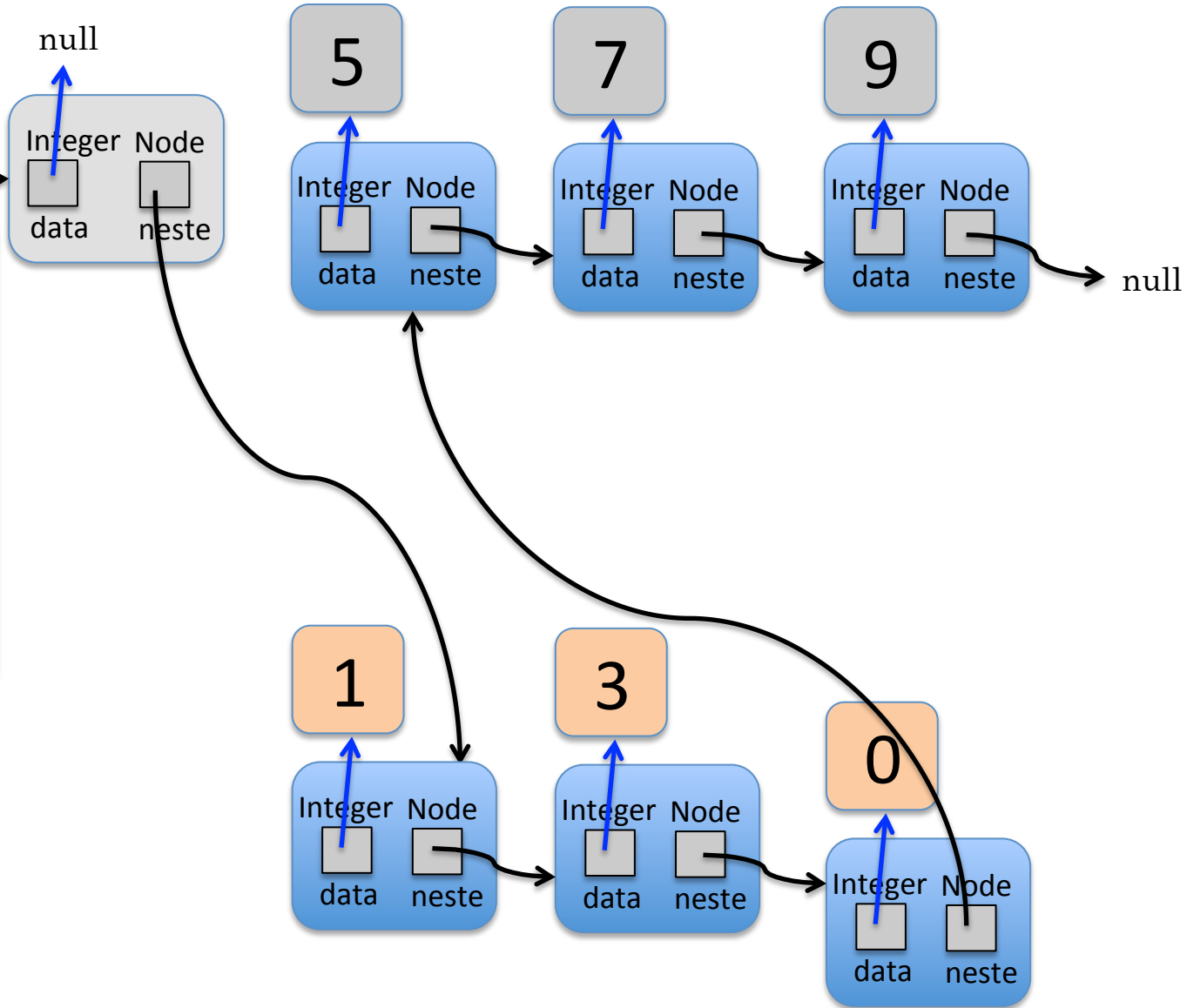
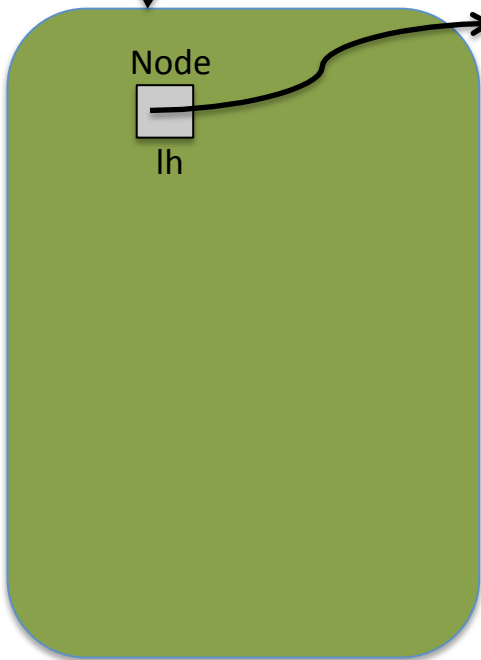
3



0



Lenkeliste<Integer>



Lenkeliste<Integer>



Node  
lh

```
private void limSammenForan ( Lenkeliste<Integer> ll ) {
  if ( ! ll.tom() ) {
    ll.bak.neste = lh.neste;
    lh.neste = ll.lh.neste;
  }
}
```

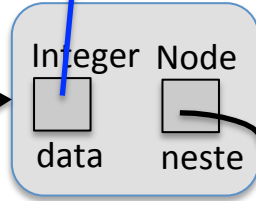
Lenkeliste<Integer>



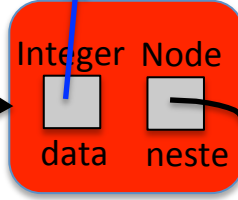
Node  
lh

Node  
bak

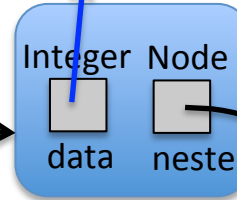
null



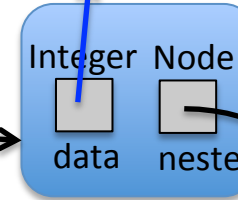
5



7



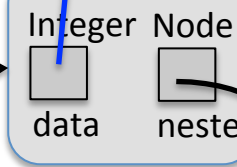
9



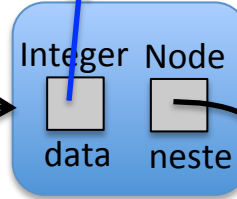
null

```
private void limSammenForan ( Lenkeliste<T> ll ) {
  if ( ! ll.tom() ) {
    ll.bak.neste = lh.neste;
    lh.neste = ll.lh.neste;
  }
}
```

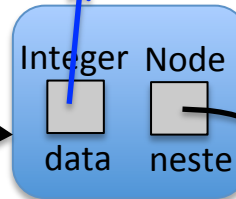
null



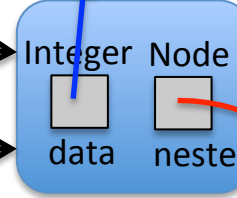
1



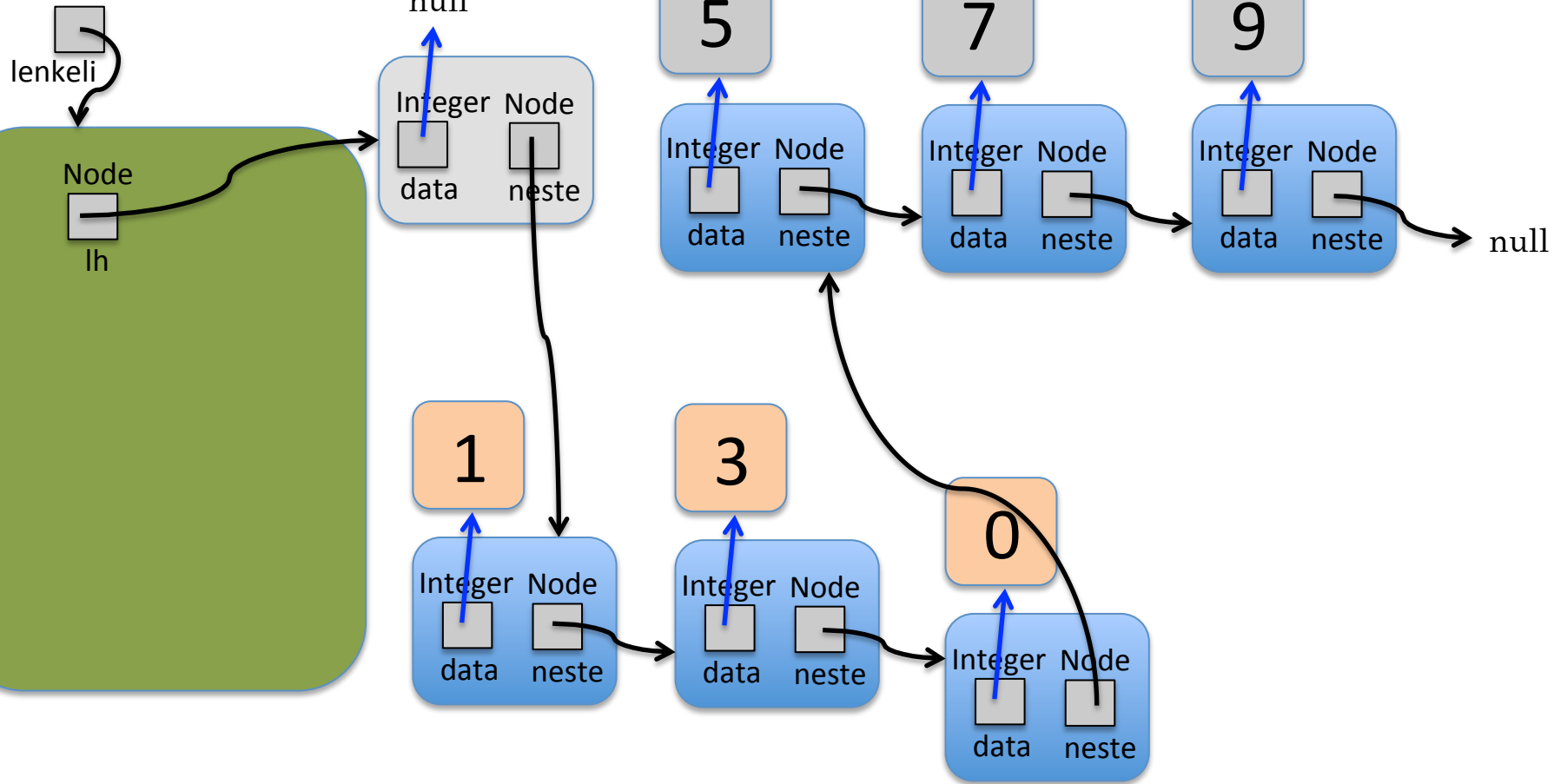
3



0



Lenkeliste<Integer>



```
private void limSammenForan ( Lenkeliste<T> l1 ) {  
    if ( ! l1.tom() ) {  
        l1.bak.neste = lh.neste;  
        lh.neste = l1.lh.neste;  
    }  
}
```

5

7

0

3

9

6

1

8

4

2

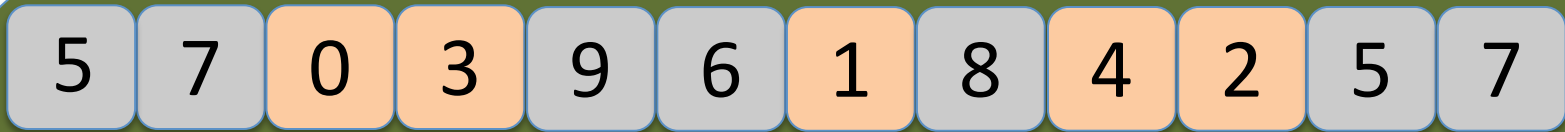
5

7

5 7 0 3 9 6 1 8 4 2 5 7

```
T pivot = this.taUtForan();  
Lenkeliste<T> mep = mindreEnnPivot(pivot);  
this.settInnForan(pivot);  
TimSammenForan(mep);
```



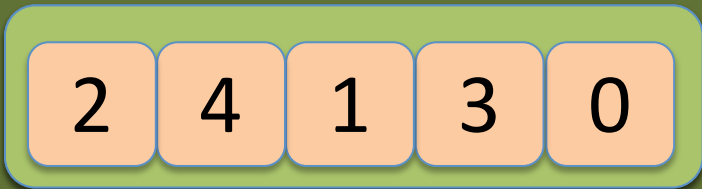
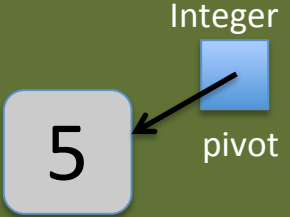


```
T pivot = this.taUtForan();  
Lenkeliste<T> mep = mindreEnnPivot(pivot);  
this.settInnForan(pivot);  
TimSammenForan(mep);
```



små verdier

store verdier



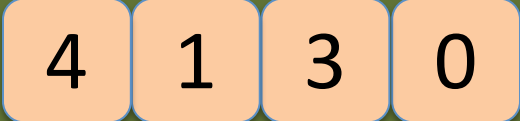


Lenkeliste<Integer>



mep





Lenkliste<Integer>



mep

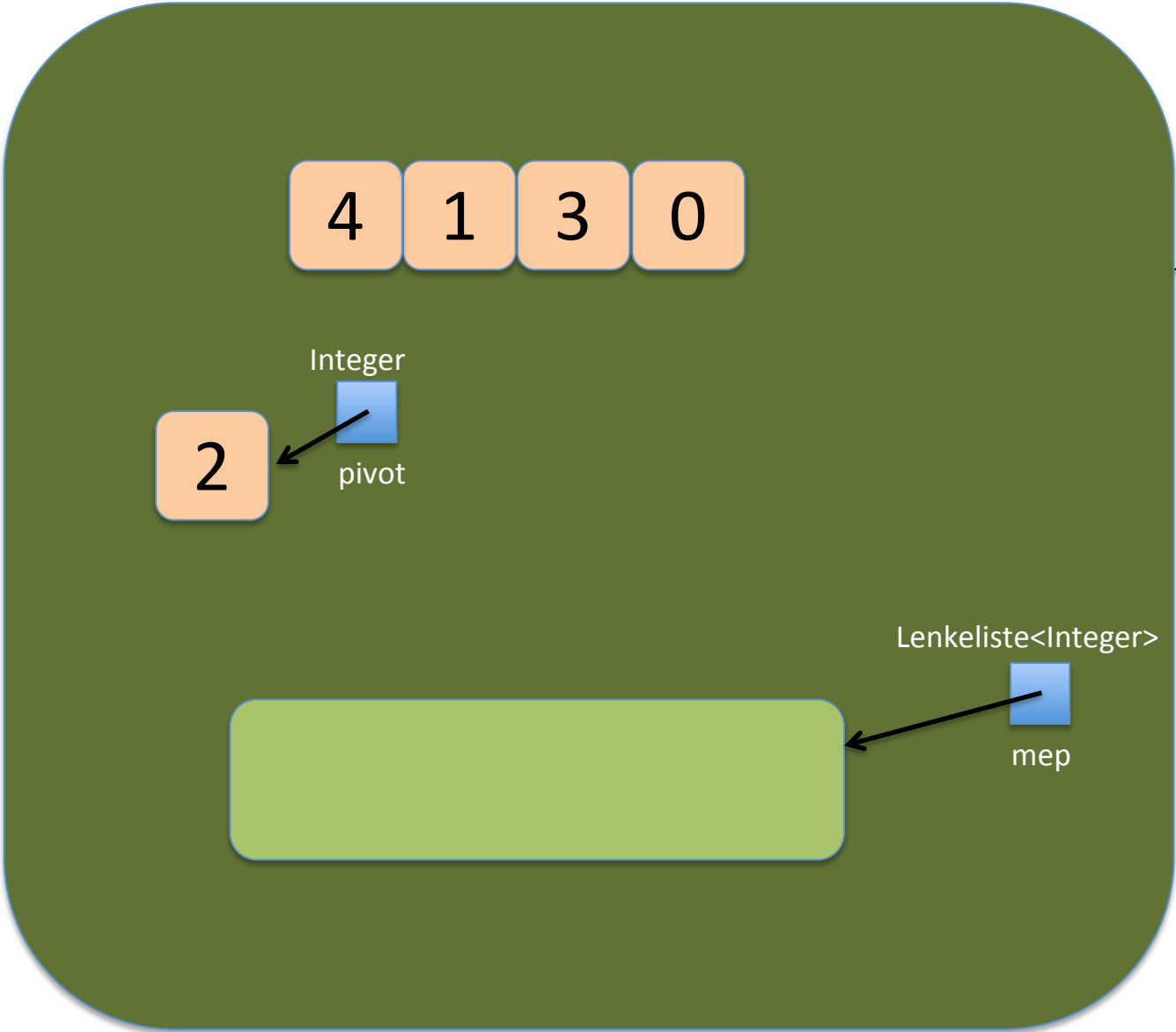


Integer



pivot

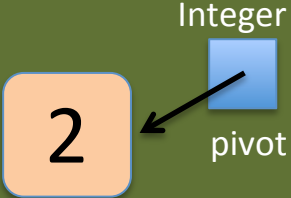
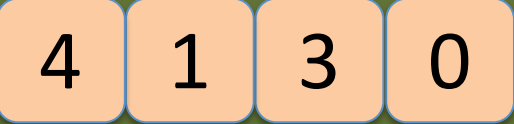




Lenkeliste<Integer>



mep

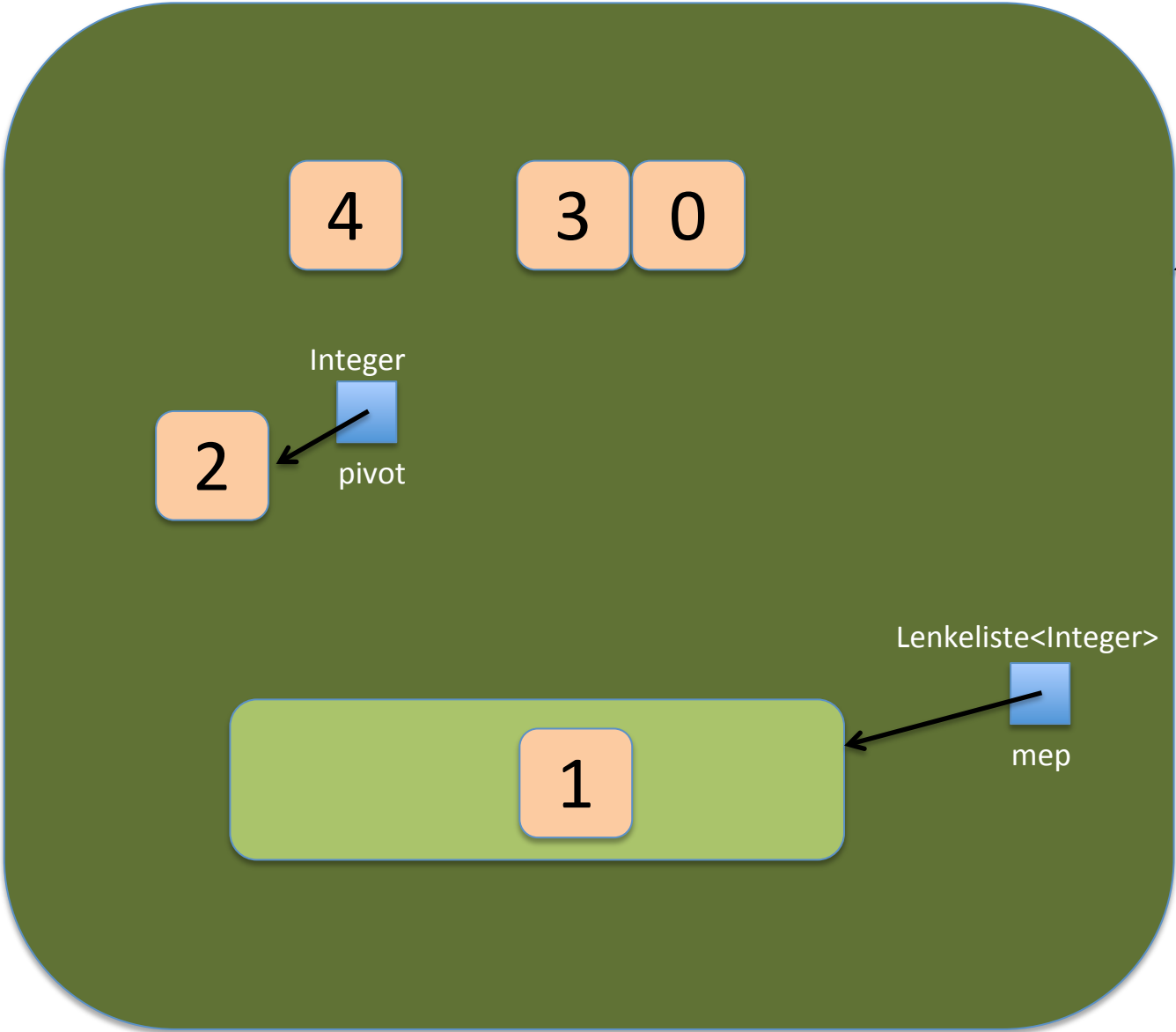


Lenkeliste<Integer>



mep





Lenkeliste<Integer>



mep

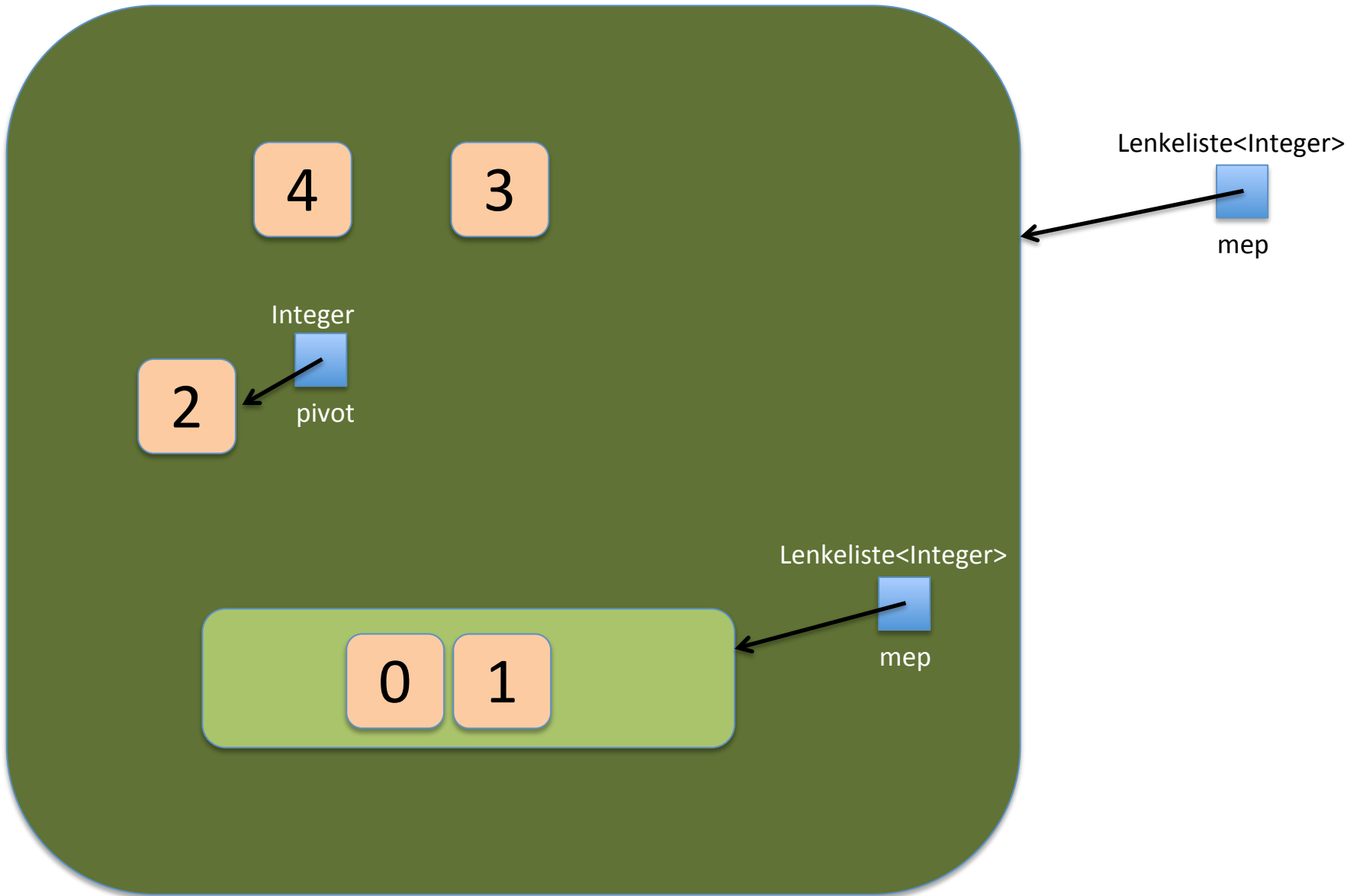


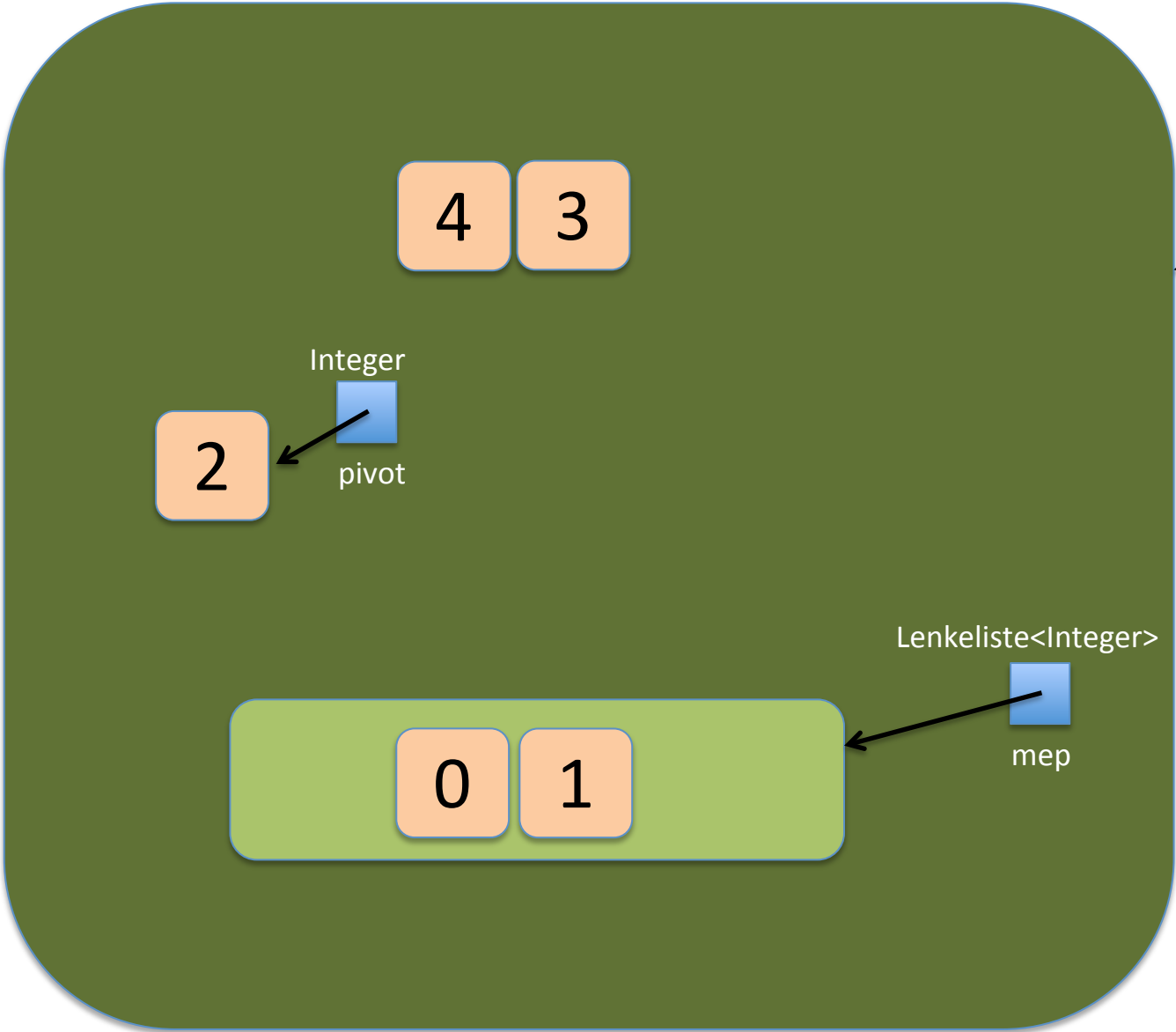
Lenkeliste<Integer>



mep







4 3

Integer  
pivot  
2

0 1

Lenkeliste<Integer>



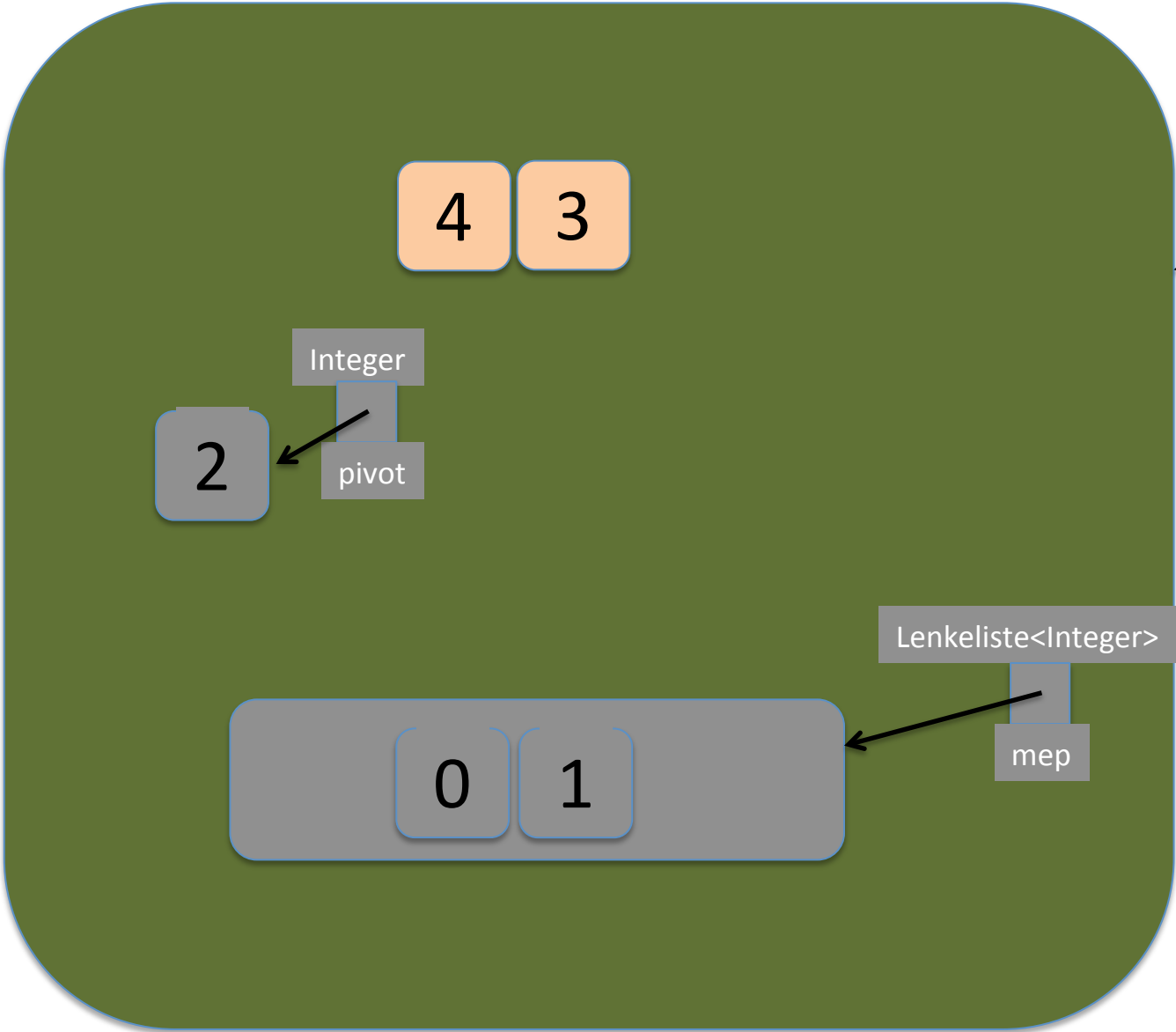
mep

Lenkeliste<Integer>



mep





Lenkeliste<Integer>



mep

Integer

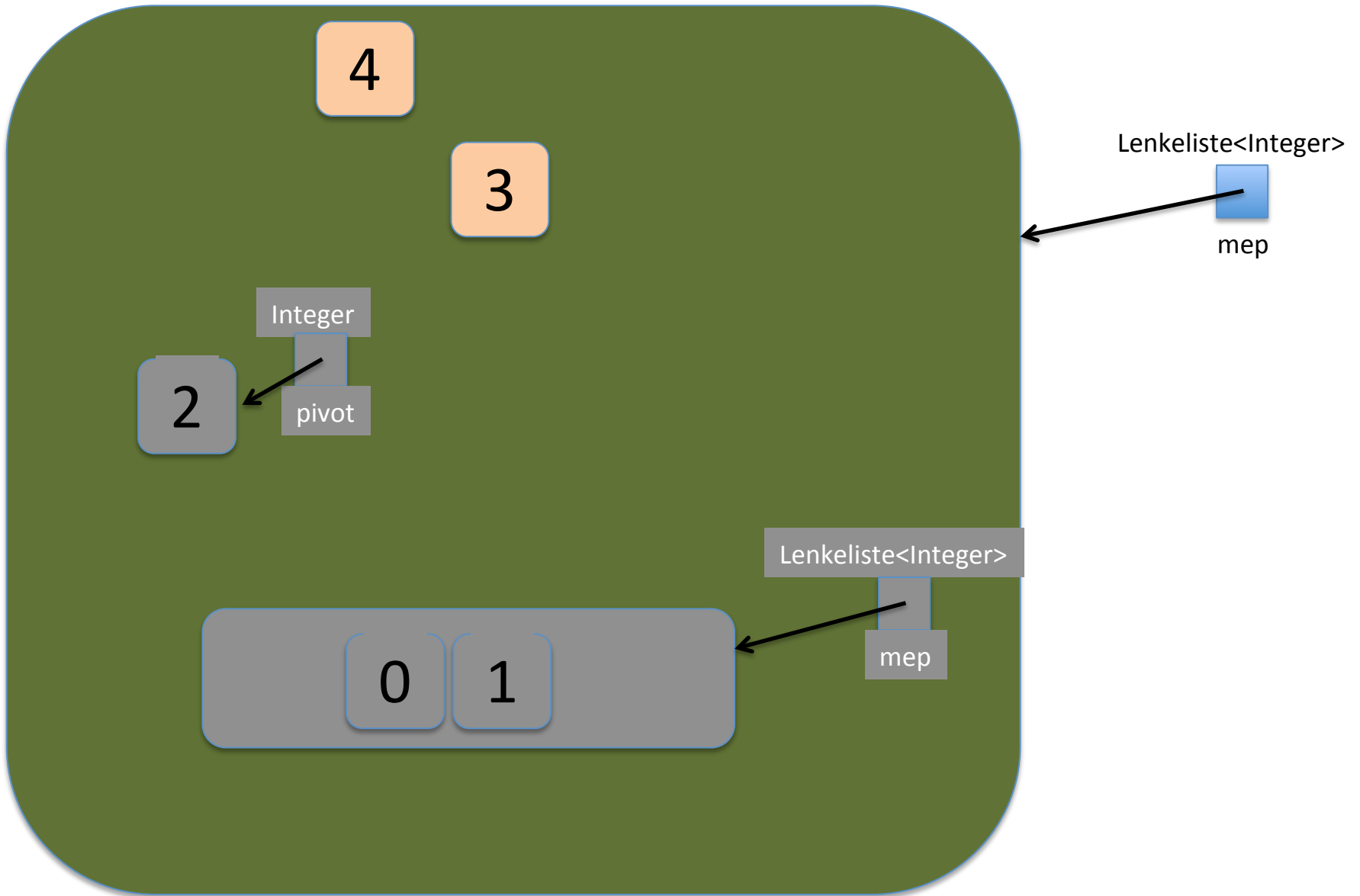
pivot

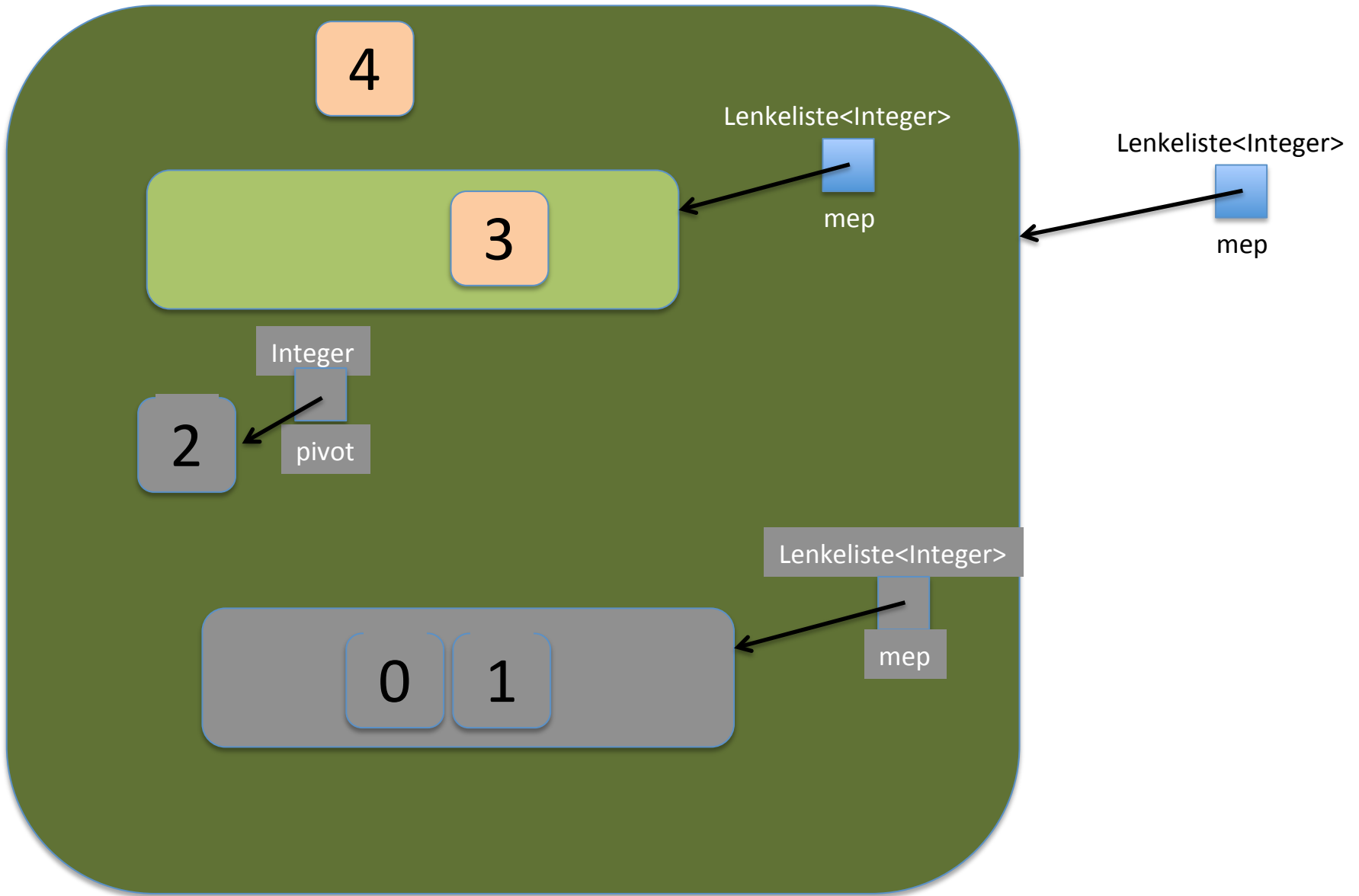


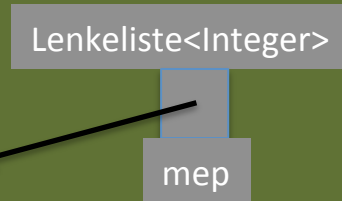
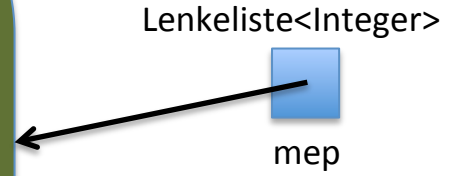
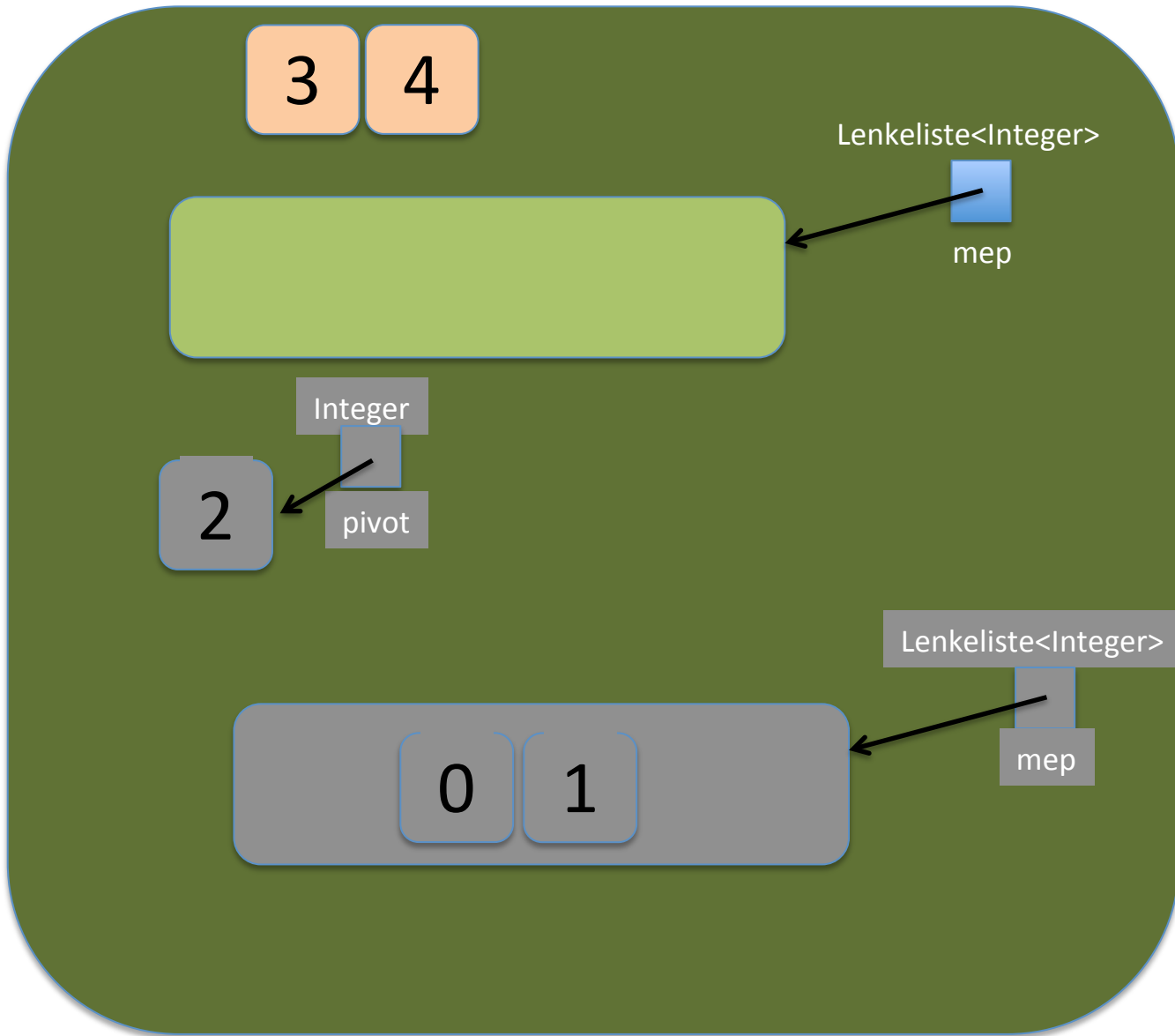
Lenkeliste<Integer>

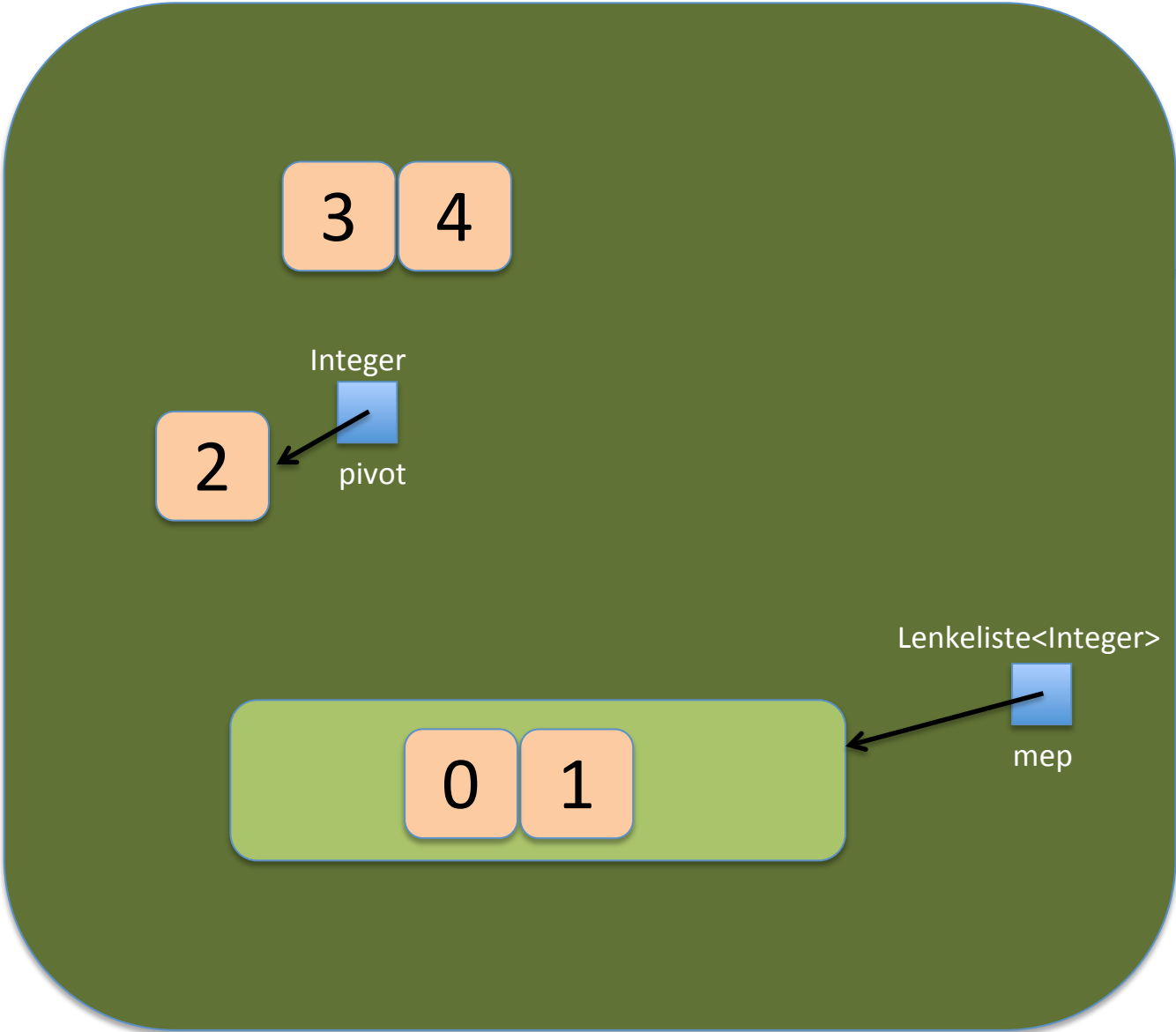
mep

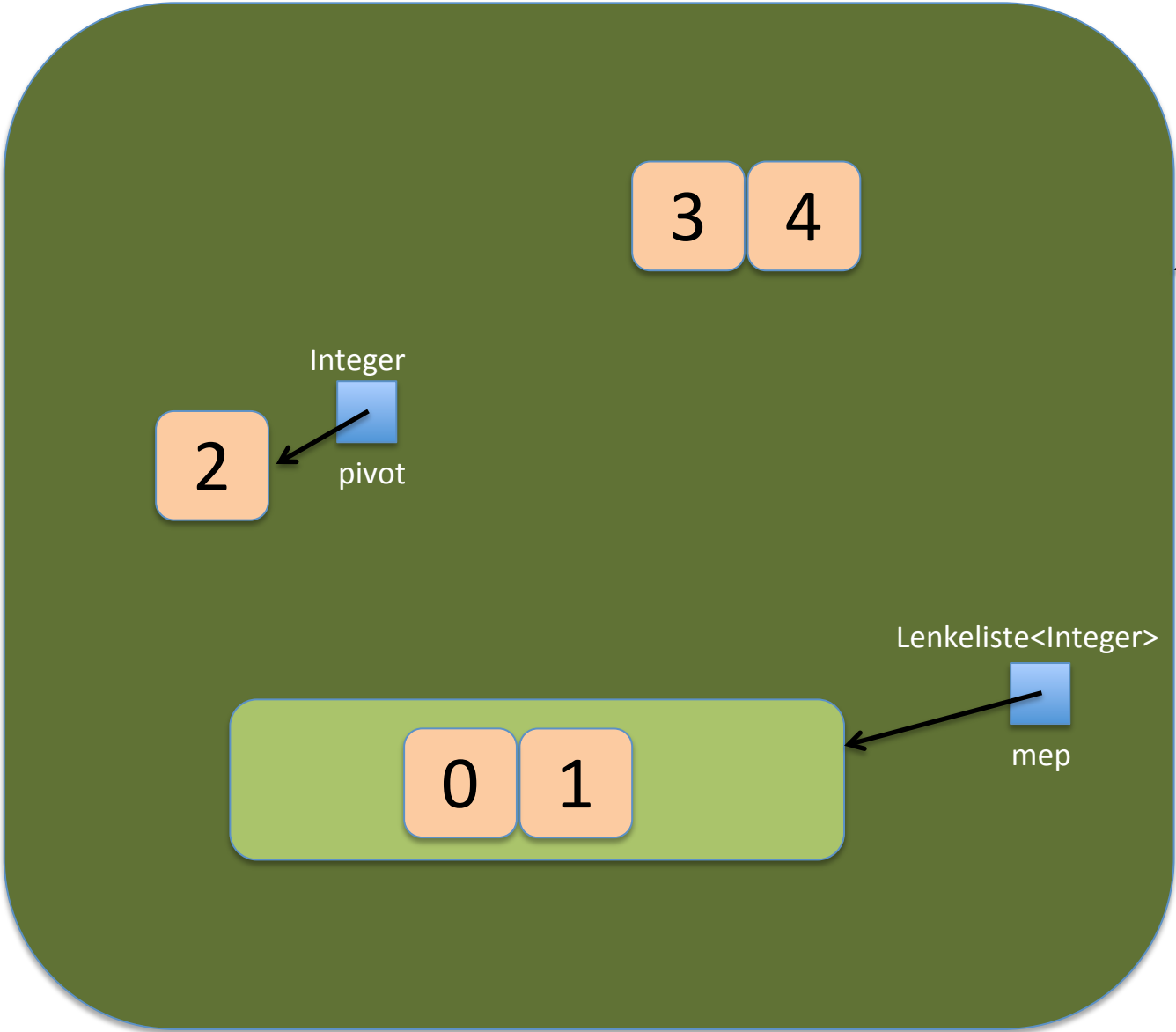












Lenkeliste<Integer>

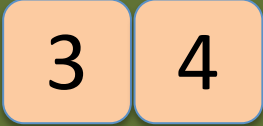


mep

Integer



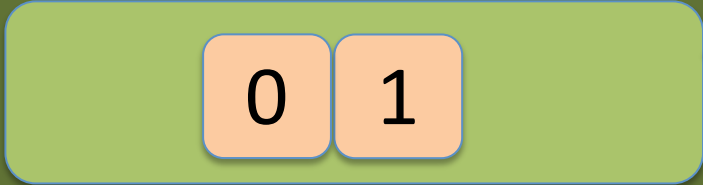
pivot

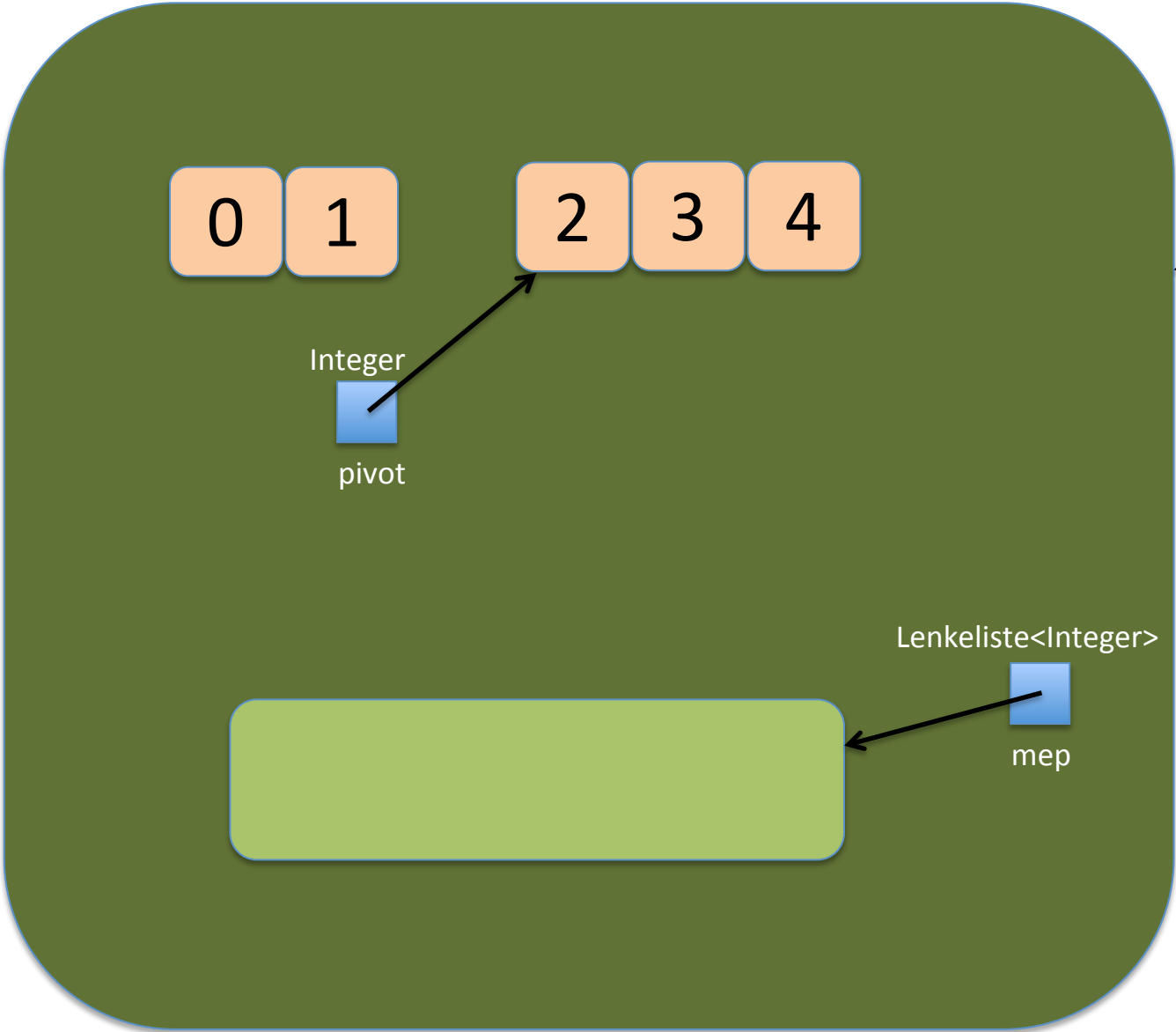


Lenkeliste<Integer>



mep





Lenkeliste<Integer>



mep

Integer

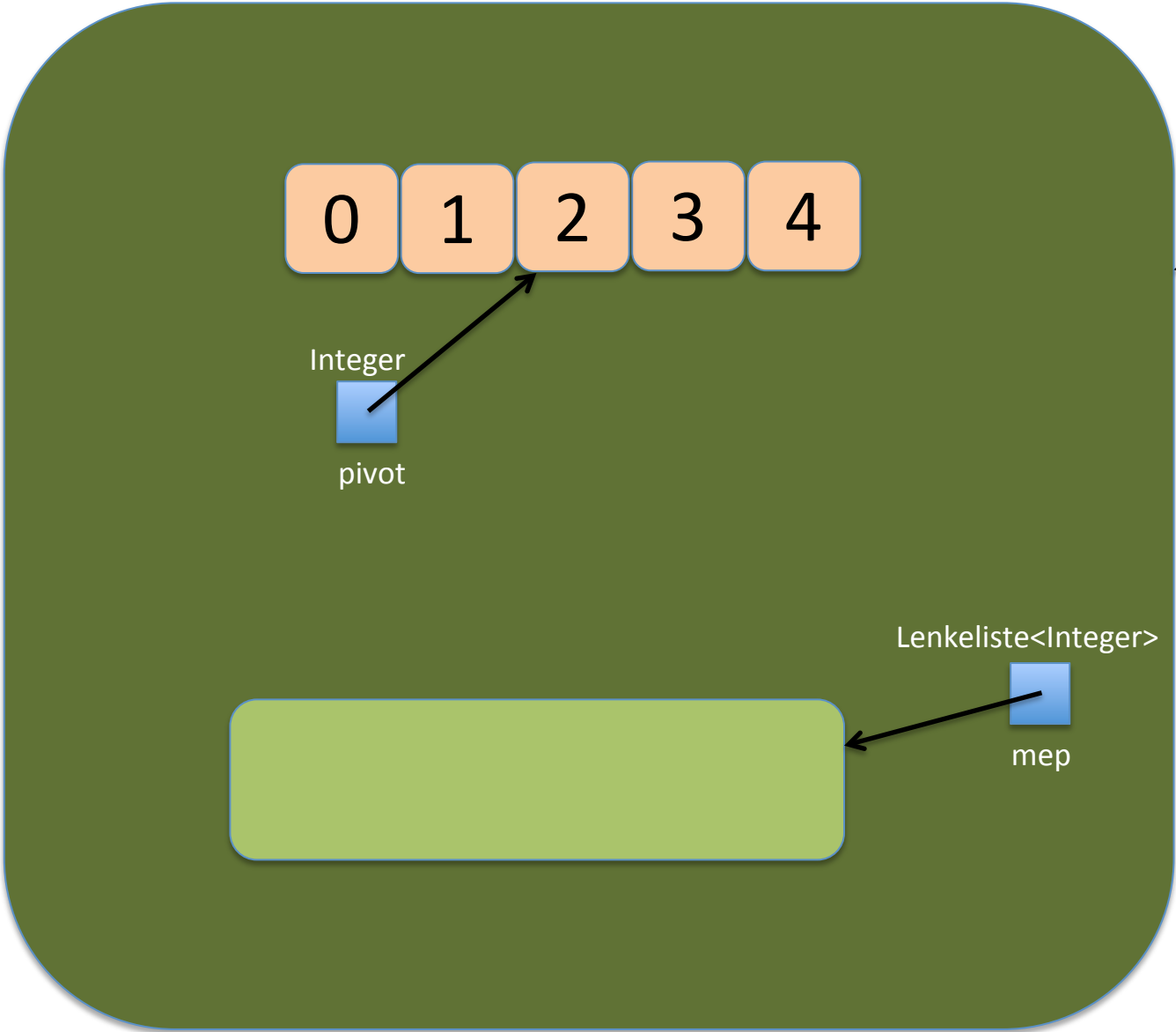


pivot

Lenkeliste<Integer>



mep



Lenkeliste<Integer>



mep

0

1

2

3

4

Integer



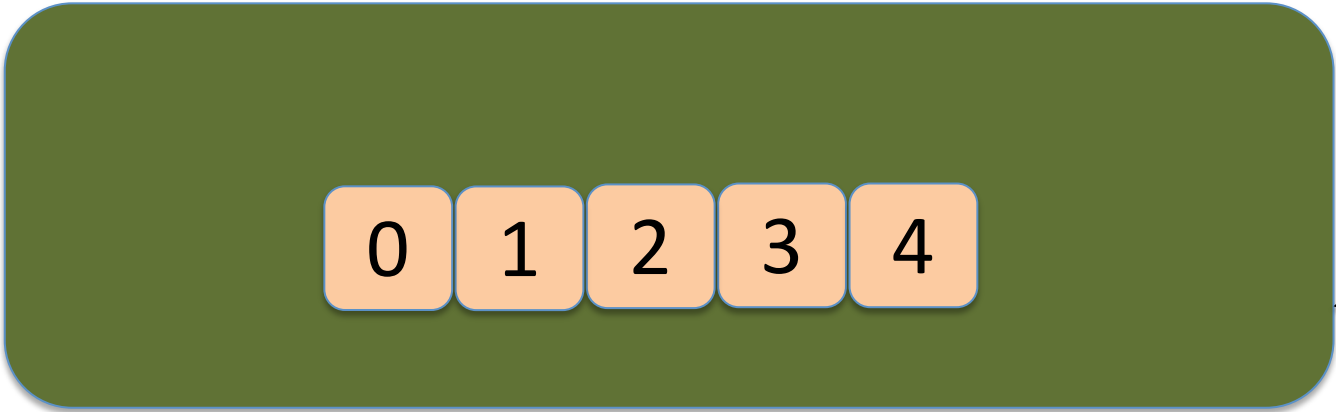
pivot

Lenkeliste<Integer>



mep



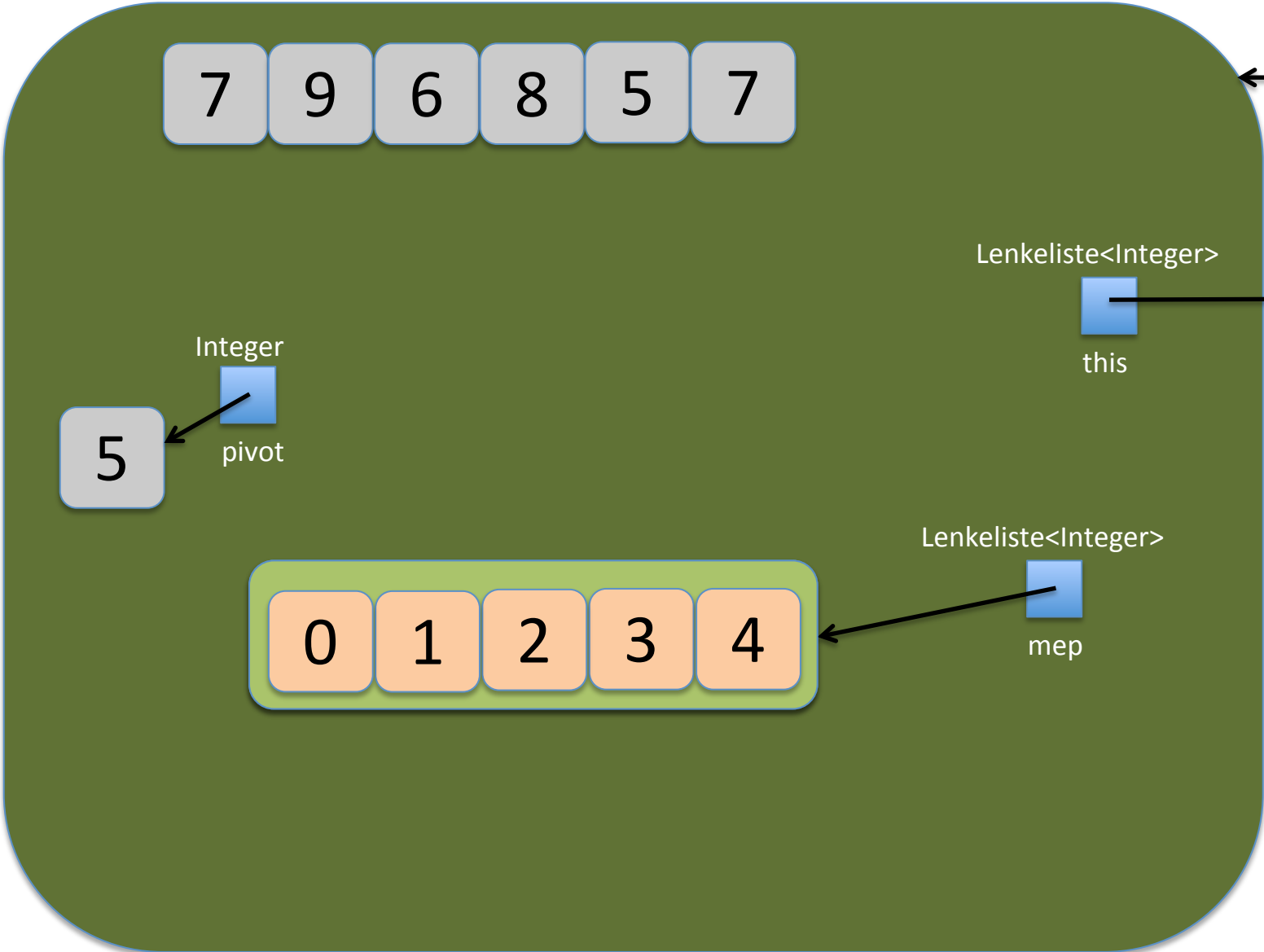


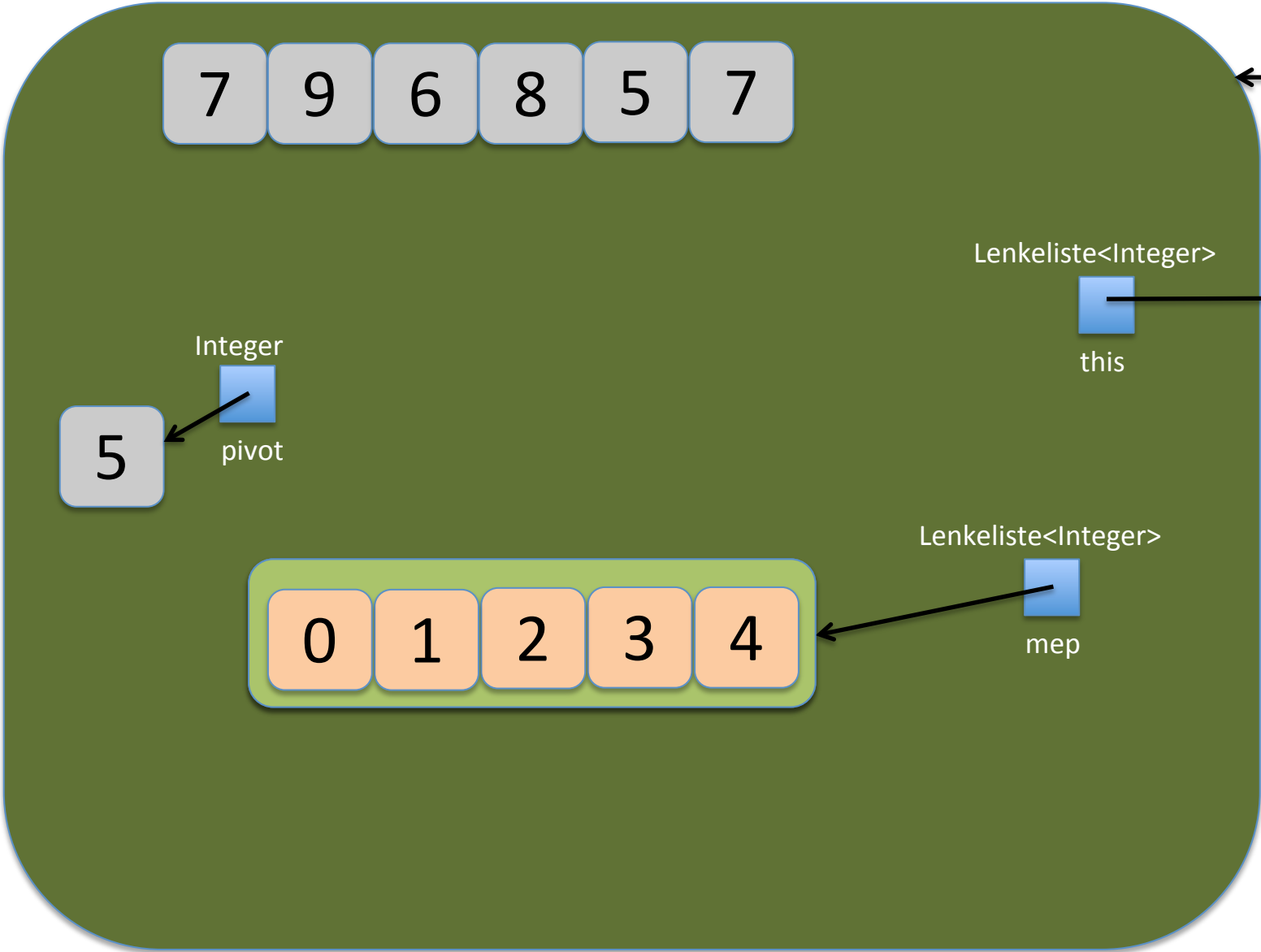
Lenkeliste<Integer>



mep







7

9

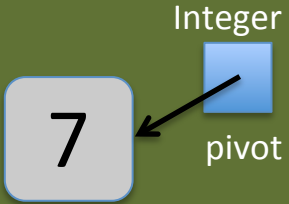
6

8

5

7

9 6 8 5 7

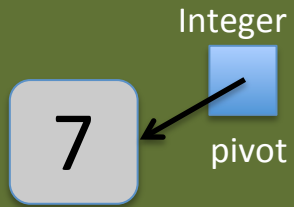


9 8 7

Integer  
pivot  
7

5 6

7 8 9



5 6



Integer  
pivot

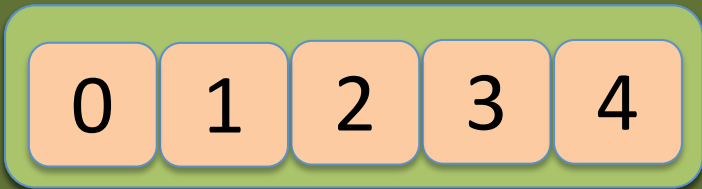
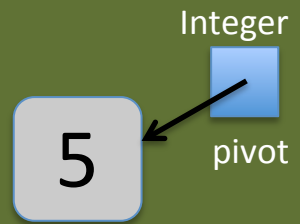
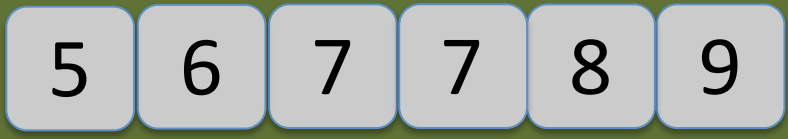


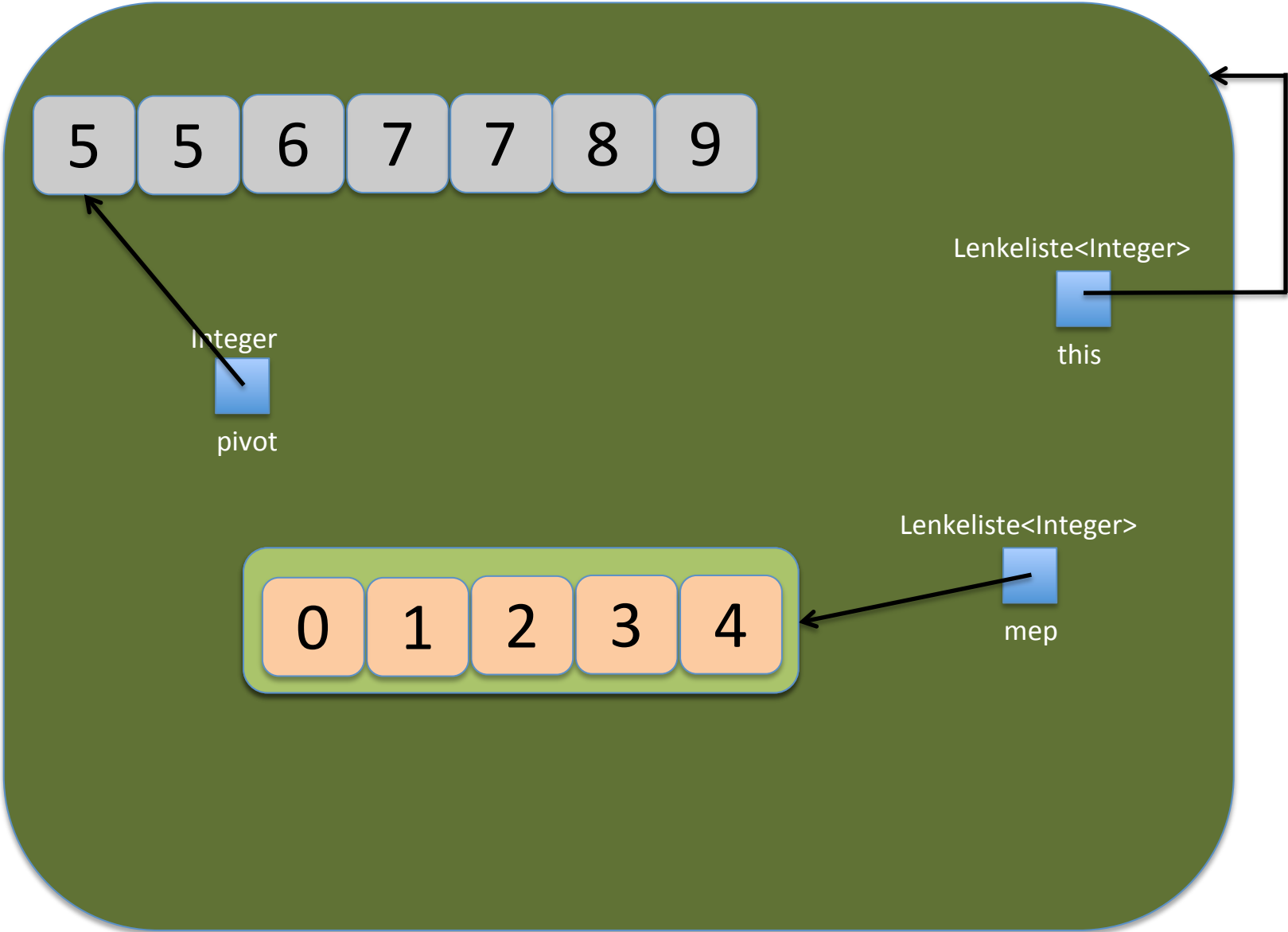


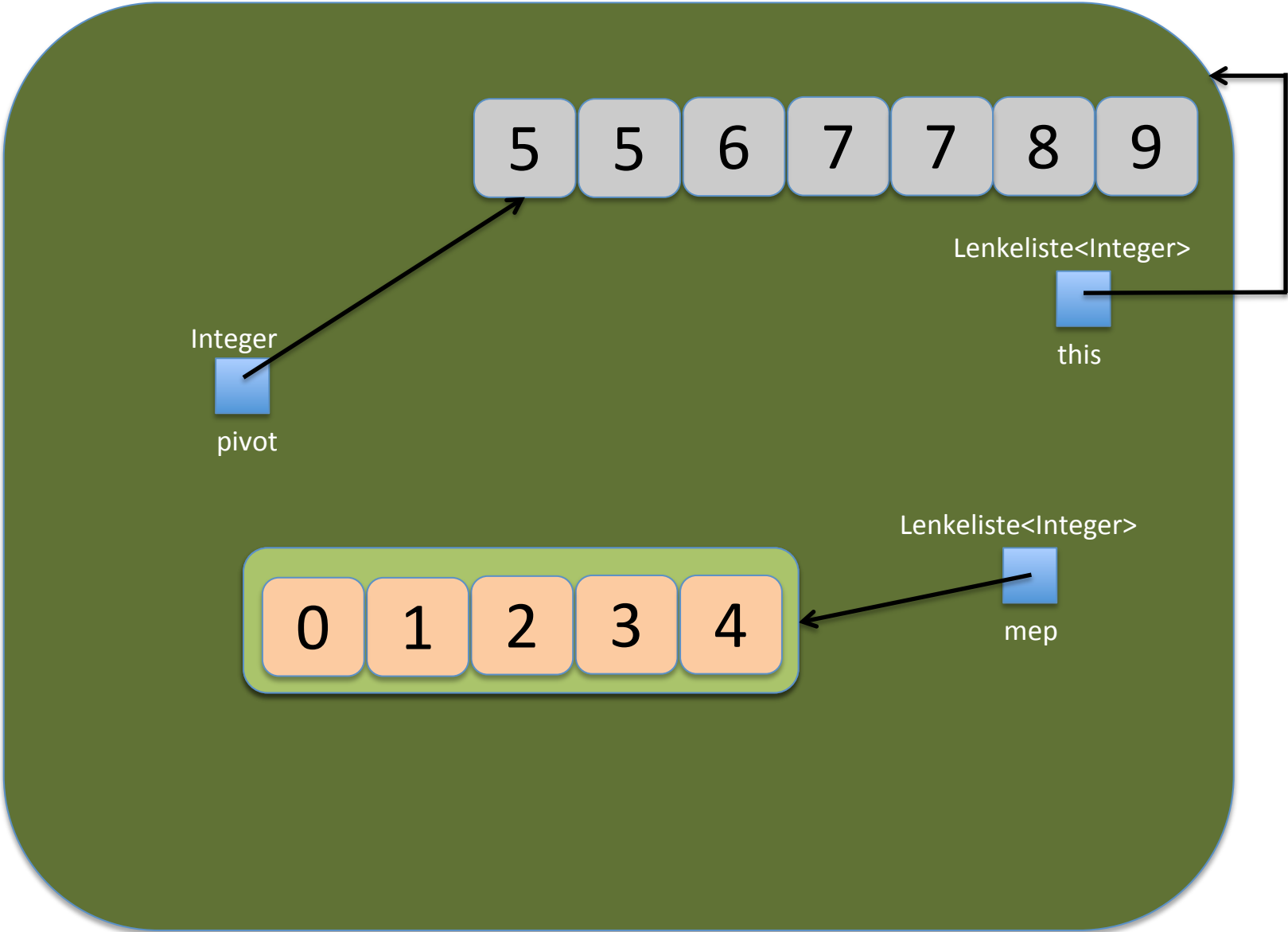


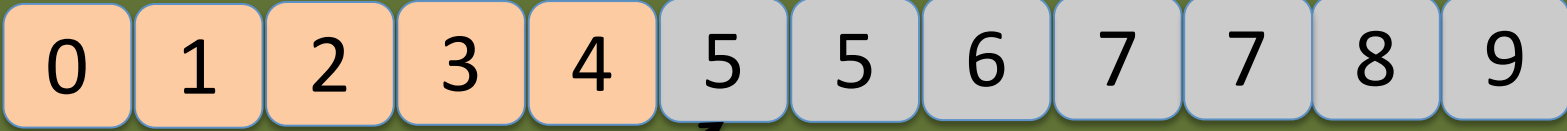
Integer  
pivot







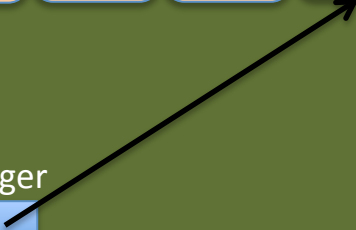




Integer



pivot





```
Lenkeliste<T> mep = new Lenkeliste<T>();  
T pivot = this.taUtForan();  
mep = mindreEnnPivot(pivot);
```



Lenkeliste<Integer>



this

```
Lenkeliste<T> mep = new Lenkeliste<T>();  
T pivot = this.taUtForan();  
mep = mindreEnnPivot(pivot);
```

Integer



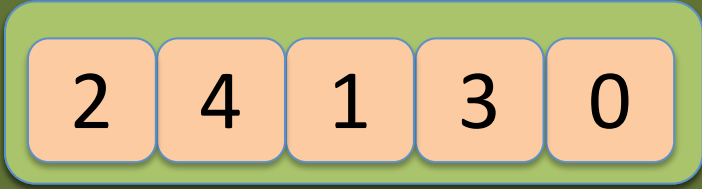
pivot



Lenkeliste<Integer>



mep





Lenkeliste<Integer>



this

```
Lenkeliste<T> mep = new Lenkeliste<T>();  
T pivot = this.taUtForan();  
mep = mindreEnnPivot(pivot);  
this.sorter();
```

Integer



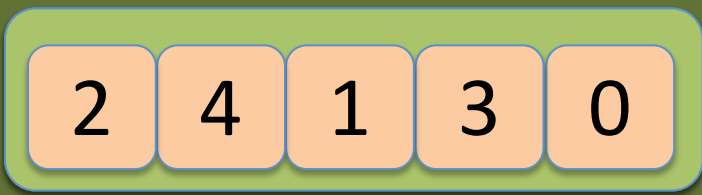
pivot



Lenkeliste<Integer>



mep







Lenkeliste<Integer>



this

```
Lenkeliste<T> mep = new Lenkeliste<T>();  
T pivot = this.taUtForan();  
mep = mindreEnnPivot(pivot);  
this.sorter();  
mep.sorter();
```

Integer



pivot

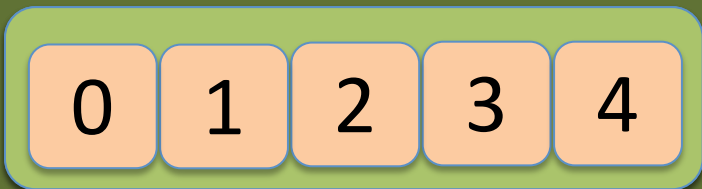


5

Lenkeliste<Integer>



mep



0

1

2

3

4



Lenkeliste<Integer>



this

Integer



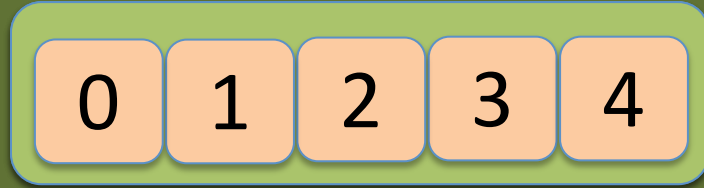
pivot

```
Lenkeliste<T> mep = new Lenkeliste<T>();  
T pivot = this.taUtForan();  
mep = mindreEnnPivot(pivot);  
this.sorter();  
mep.sorter();  
this.settInnForan(pivot);
```

Lenkeliste<Integer>



mep





Lenkeliste<Integer>



this

Integer



pivot

```
Lenkeliste<T> mep = new Lenkeliste<T>();  
T pivot = this.taUtForan();  
mep = mindreEnnPivot(pivot);  
this.sorter();  
mep.sorter();  
this.settInnForan(pivot);  
TimSammenForan(mep);
```



```
Lenkeliste<T> mep = new Lenkeliste<T>();  
T pivot = this.taUtForan();  
mep = mindreEnnPivot(pivot);  
this.sorter();  
mep.sorter();  
this.settInnForan(pivot);  
TimSammenForan(mep);
```



```
public void sorter() {  
    Lenkeliste<T> mep = new Lenkeliste<T>();  
    T pivot = this.taUtForan();  
    mep = mindreEnnPivot(pivot);  
    this.sorter();  
    mep.sorter();  
    this.settInnForan(pivot);  
    TimSammenForan(mep);  
}
```

```
public void quickSort() {
```

```
    if (! tom() ) {  
        Lenkeliste<T> mep = new Lenkeliste<T>();  
        T pivot = this.taUtForan();  
        mep = mindreEnnPivot(pivot);  
        mep.quickSort(); this.quickSort();  
        this.settInnForan(pivot);  
        TimSammenForan(mep);  
    }
```

```
}
```

5 7 0 3 9 6 1 8 4 2 5 7

```
public void quickSort() {
```

```
    if (! tom() ) {  
        Lenkeliste<T> mep = new Lenkeliste<T>();  
        T pivot = this.taUtForan();  
        mep = mindreEnnPivot(pivot);  
        mep.quickSort(); this.quickSort();  
        this.settInnForan(pivot);  
        TimSammenForan(mep);  
    }
```

```
}
```

5

7

0

3

9

6

1

8

4

2

5

7

```
public void quickSort() {
```

```
    if (! tom() ) {  
        Lenkeliste<T> mep = new Lenkeliste<T>();  
        T pivot = this.taUtForan();  
        mep = mindreEnnPivot(pivot);  
        mep.quickSort(); this.quickSort();  
        this.settInnForan(pivot);  
        TimSammenForan(mep);  
    }
```

```
}
```



5

2

4

1

3

0

7

9

6

8

5

7

```
public void quickSort() {
```

```
    if (! tom() ) {  
        Lenkeliste<T> mep = new Lenkeliste<T>();  
        T pivot = this.taUtForan();  
        mep = mindreEnnPivot(pivot);  
        mep.quickSort();  
        this.quickSort();  
        this.settInnForan(pivot);  
        TimSammenForan(mep);  
    }  
}
```

5

0

1

2

3

4

7

9

6

8

5

7

```
public void quickSort() {
```

```
    if (! tom() ) {  
        Lenkeliste<T> mep = new Lenkeliste<T>();  
        T pivot = this.taUtForan();  
        mep = mindreEnnPivot(pivot);  
        mep.quickSort();  
        this.quickSort();  
        this.settInnForan(pivot);  
        TimSammenForan(mep);  
    }
```

```
}
```

5

0

1

2

3

4

7

9

6

8

5

7

```
public void quickSort() {
```

```
    if (! tom() ) {  
        Lenkeliste<T> mep = new Lenkeliste<T>();  
        T pivot = this.taUtForan();  
        mep = mindreEnnPivot(pivot);  
        mep.quickSort();  
        this.quickSort();  
        this.settInnForan(pivot);  
        TimSammenForan(mep);  
    }
```

```
}
```

0 1 2 3 4 5 7 9 6 8 5 7

```
public void quickSort() {
```

```
    if (! tom() ) {  
        Lenkeliste<T> mep = new Lenkeliste<T>();  
        T pivot = this.taUtForan();  
        mep = mindreEnnPivot(pivot);  
        mep.quickSort();  
        this.quickSort();  
        this.settInnForan(pivot);  
        TimSammenForan(mep);  
    }
```

```
}
```

0 1 2 3 4 5 7 9 6 8 5 7

```
public void quickSort() {
```

```
    if (! tom() ) {  
        Lenkeliste<T> mep = new Lenkeliste<T>();  
        T pivot = this.taUtForan();  
        mep = mindreEnnPivot(pivot);  
        mep.quickSort();  
        this.quickSort();  
        this.settInnForan(pivot);  
        TimSammenForan(mep);  
    }
```

```
}
```

0 1 2 3 4 5 7 9 6 8 5 7

```
public void quickSort() {
```

```
    if (! tom() ) {  
        Lenkeliste<T> mep = new Lenkeliste<T>();  
        T pivot = this.taUtForan();  
        mep = mindreEnnPivot(pivot);  
        mep.quickSort();  
        this.quickSort();  
        this.settInnForan(pivot);  
        TimSammenForan(mep);  
    }
```

```
}
```

0 1 2 3 4 5 7 9 6 8 5 7

et voilà !