

Informatik Q1 Abels



Sortieralgorithmen

Sortieralgorithmen



Insertionsort

Insertionsort

klein

groß

2	8	5	3	9	4
---	---	---	---	---	---

0

1

2

3

4

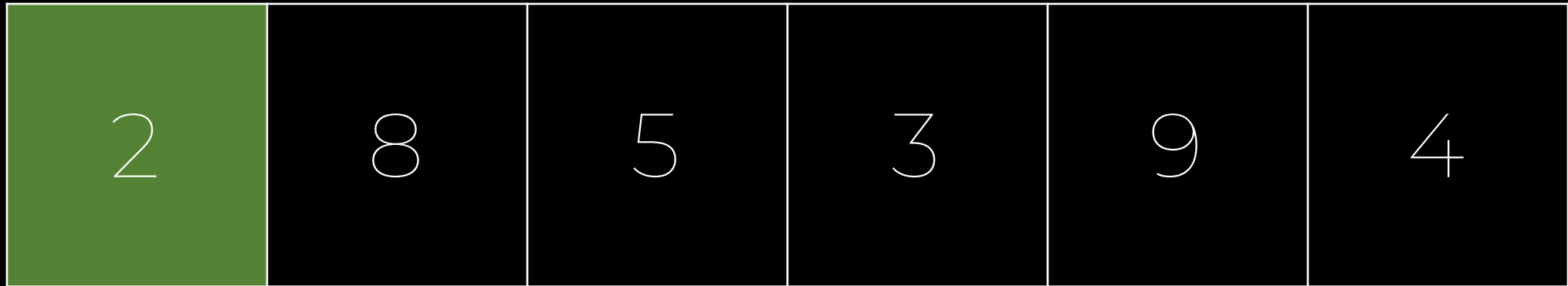
5



Insertionsort

klein

groß



0

1

2

3

4

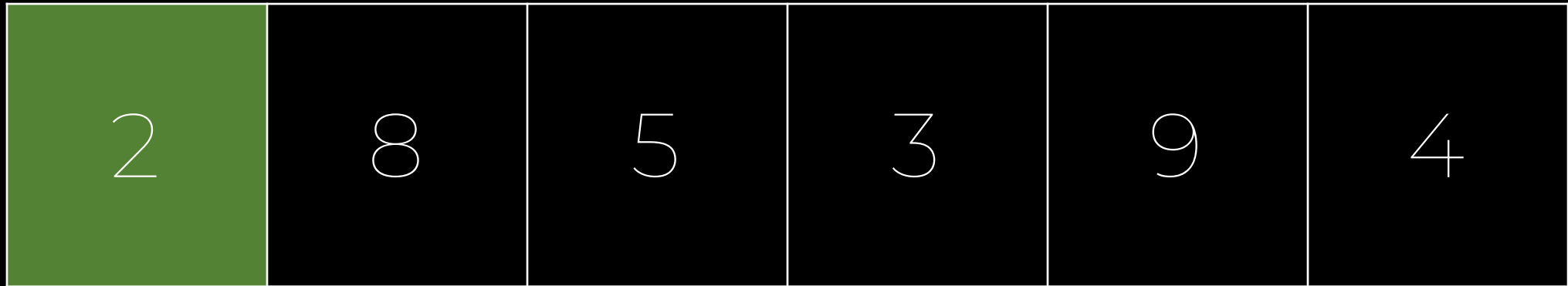
5



Insertionsort

klein

groß



0

1

2

3

4

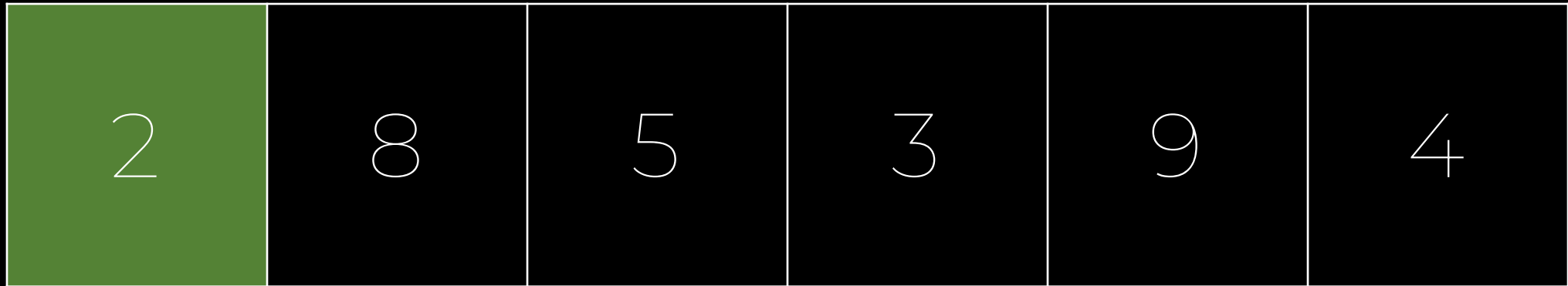
5



Insertionsort

klein

groß



0



1



2

3

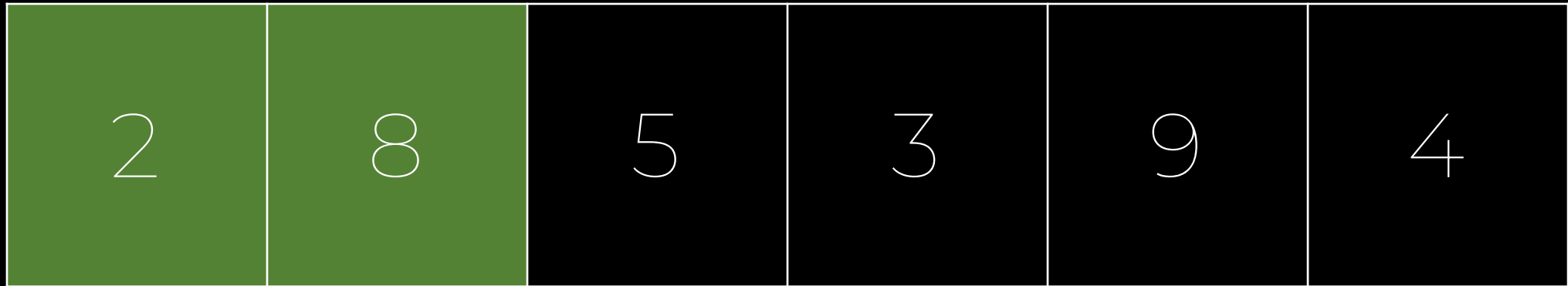
4

5

Insertionsort

klein

groß



0

1

2

3

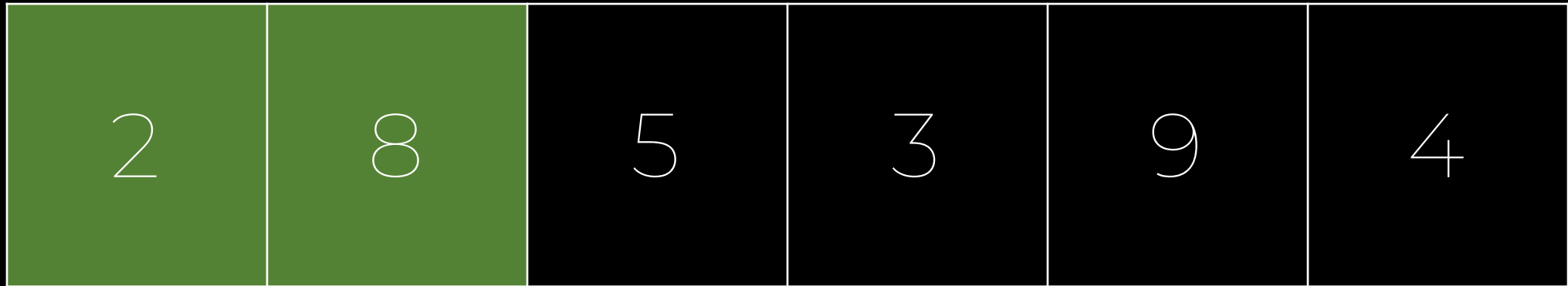
4

5

Insertionsort

klein

groß



0

1

2

3

4

5



Insertionsort

klein

groß



0

1

2

3

4

5



Insertionsort

klein

groß



0

1

2

3

4

5



Insertionsort

klein

groß



0

1

2

3

4

5

Insertionsort

klein

groß



0

1

2

3

4

5



Insertionsort

klein

groß



0

1

2

3

4

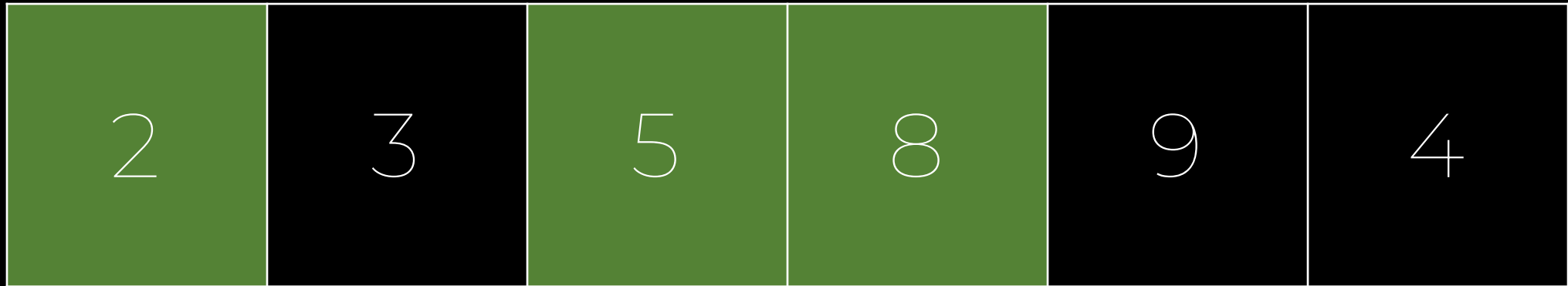
5



Insertionsort

klein

groß



0

1

2

3

4

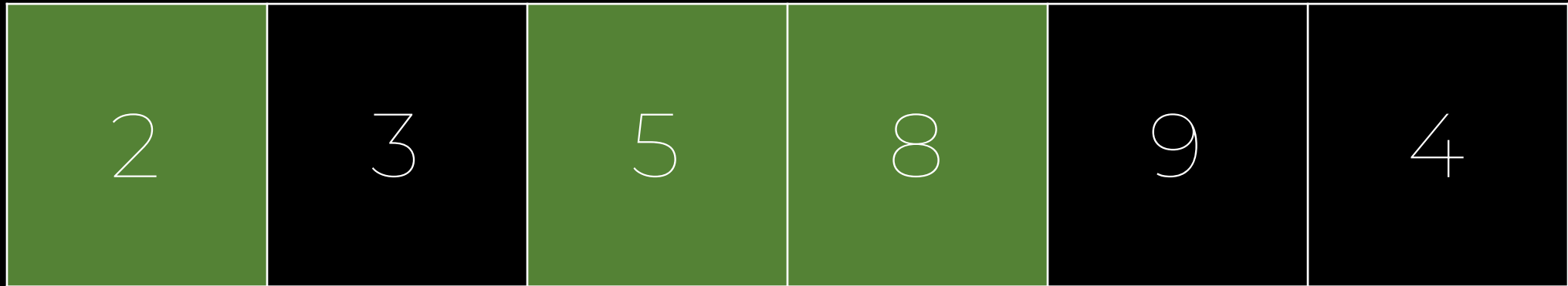
5



Insertionsort

klein

groß



0

1

2

3

4

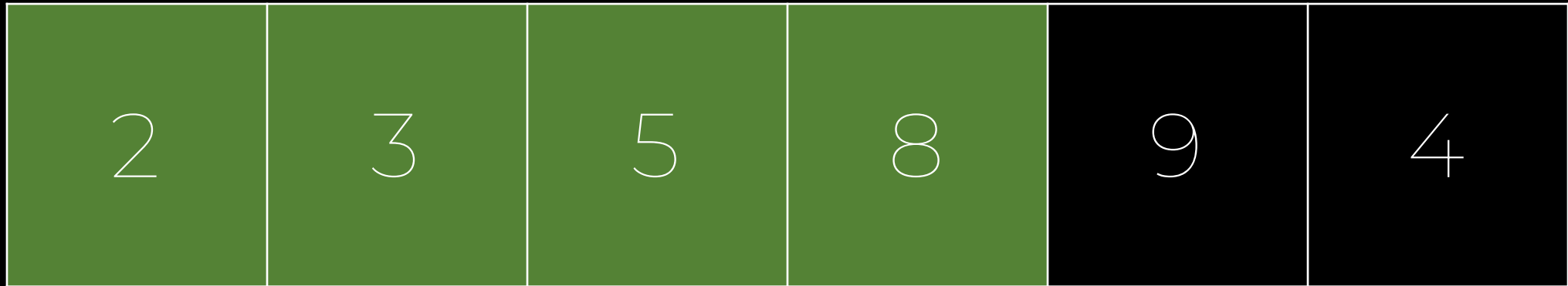
5



Insertionsort

klein

groß



0

1

2

3

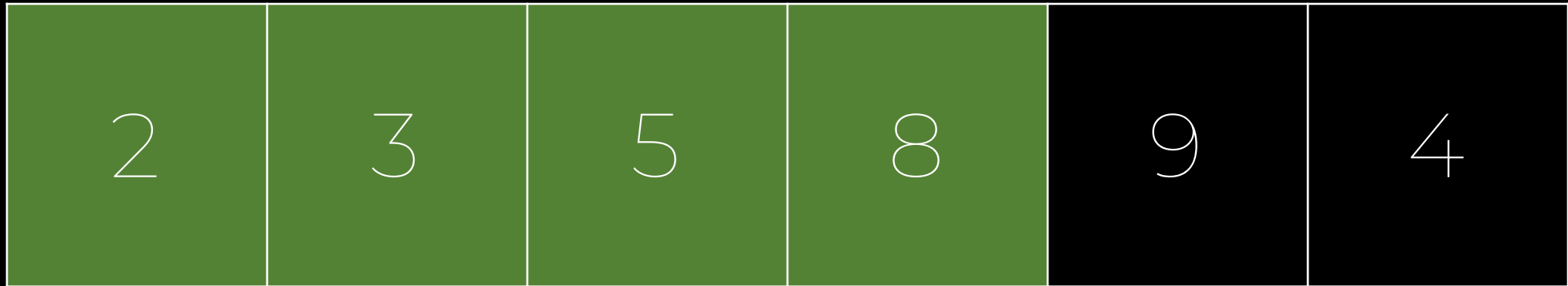
4

5

Insertionsort

klein

groß



0

1

2

3

4

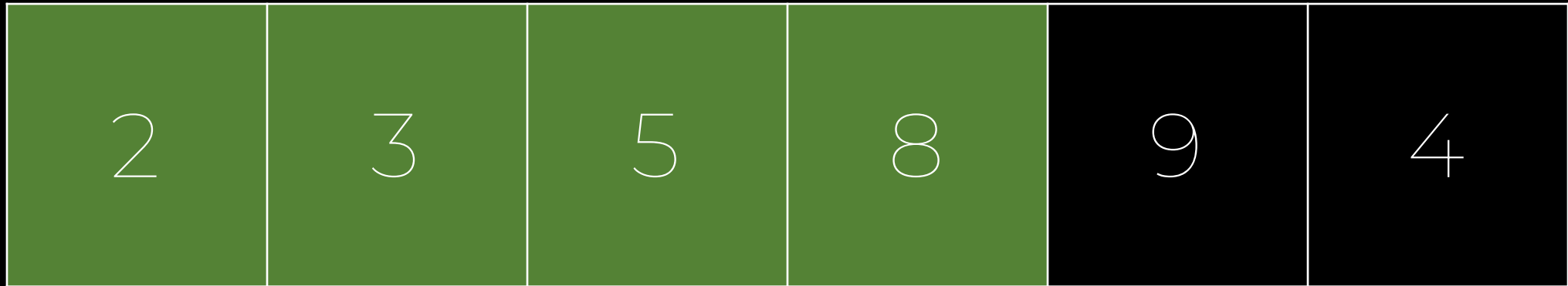
5



Insertionsort

klein

groß



0

1

2

3

4

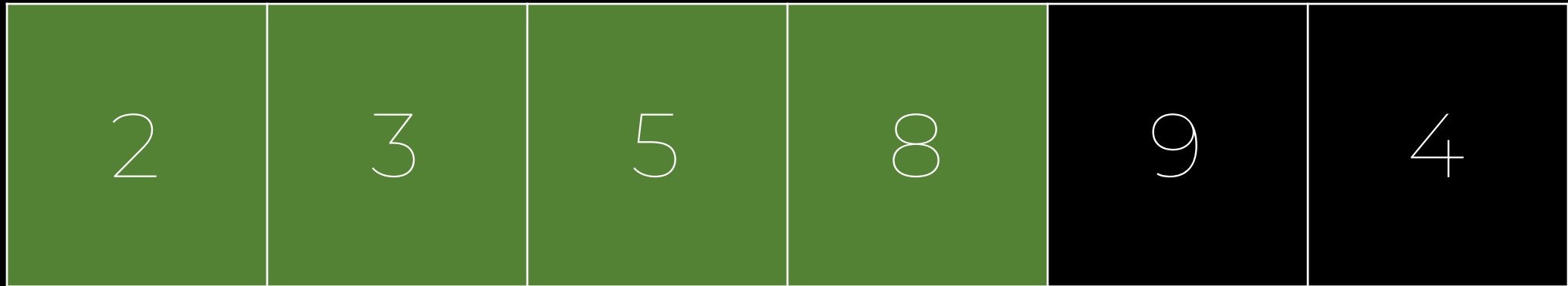
5



Insertionsort

klein

groß



0

1

2

3

4

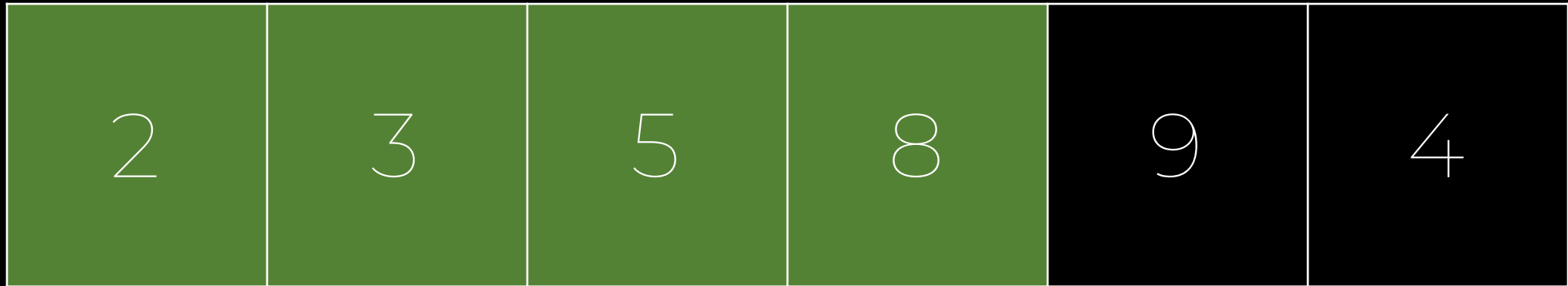
5



Insertionsort

klein

groß



0

1

2

3

4

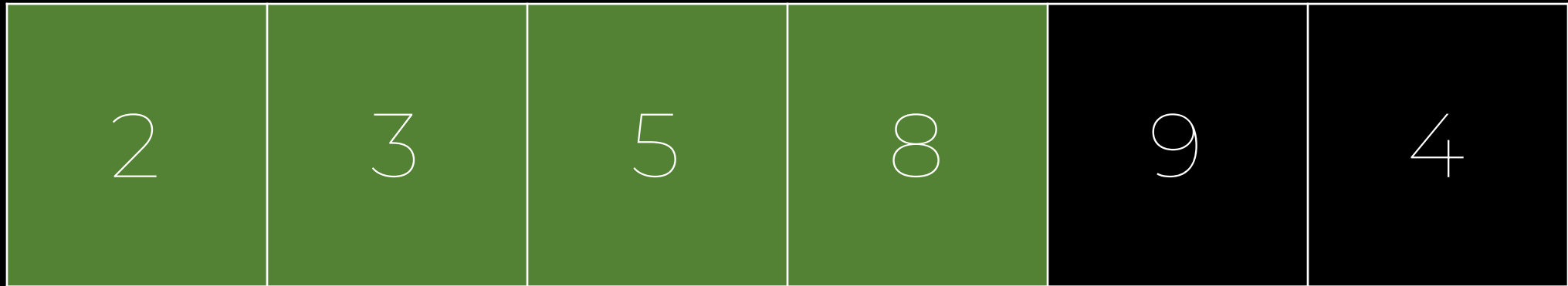
5



Insertionsort

klein

groß



0

1

2

3

4

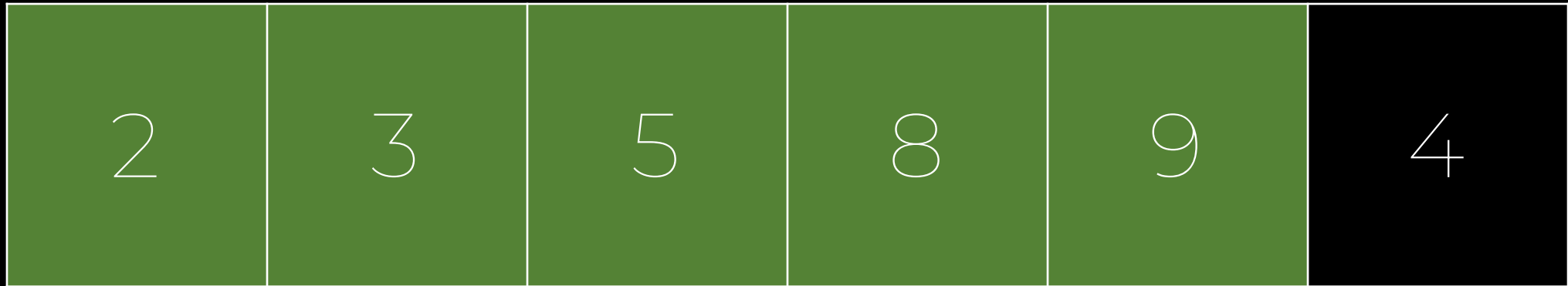
5



Insertionsort

klein

groß



0

1

2

3

4

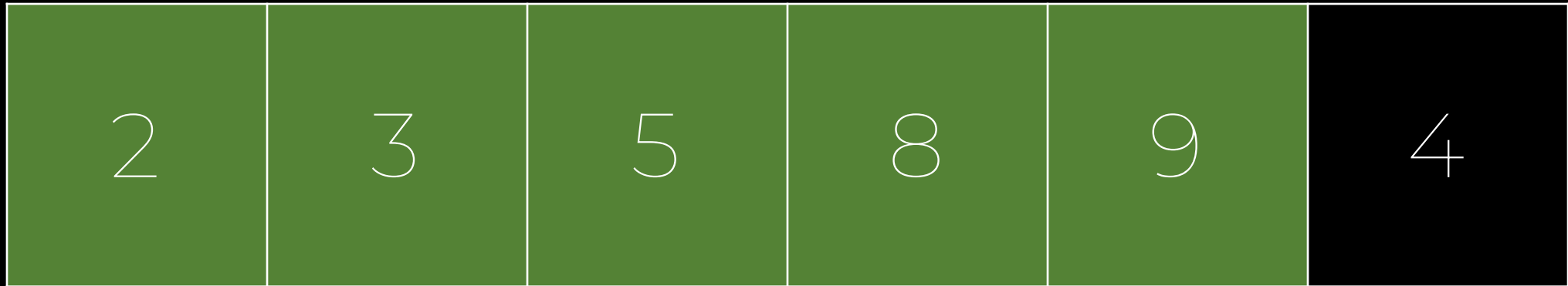
5



Insertionsort

klein

groß



0

1

2

3

4

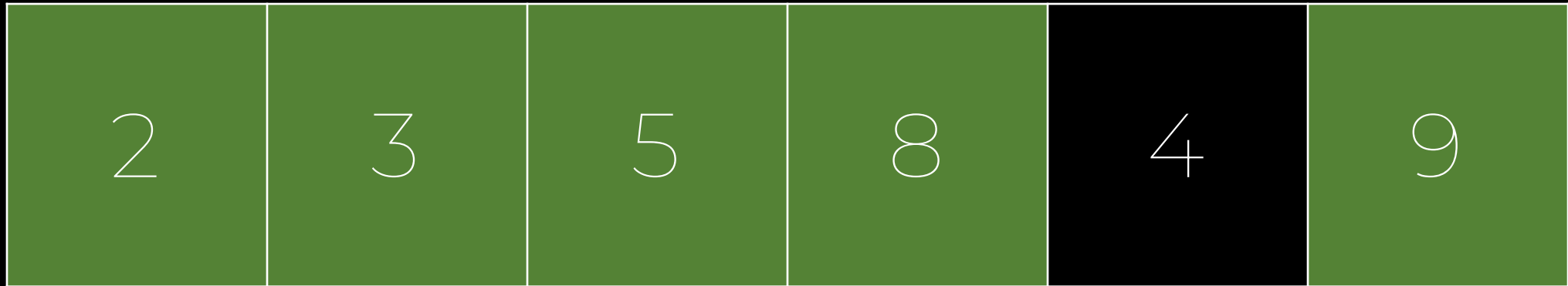
5



Insertionsort

klein

groß



0

1

2

3

4

5



Insertionsort

klein

groß



0

1

2

3

4

5

Insertionsort

klein

groß



0

1

2

3

4

5



Insertionsort

klein

groß



0

1

2

3

4

5



Insertionsort

klein

groß



0

1

2

3

4

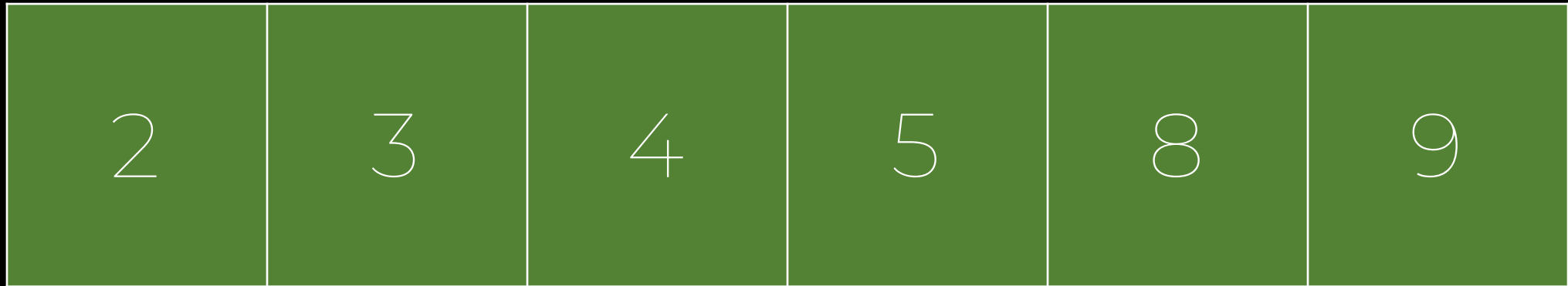
5



Insertionsort

klein

groß



0

1

2

3

4

5



Übung 1

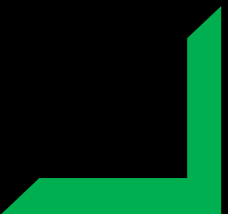
- a) Sortiere folgenden Zahlenfolge mit dem Insertionsort:

5 2 4 6 1 3

- b) Erkläre den Insertionsort in deinen eigenen Worten.
- c) Analysiere die Komplexität (Laufzeit) des Insertionsort.
- d) Zeichne ein Struktogramm zum Insertionsort.
- e) Erstelle ein Programm namens **Sortierverfahren** und implementiere den Insertionsort.

```
Sortierverfahren.py

1 def insertion_sort(liste):
2     # DEIN CODE
3     return liste
4
5 print("Testcase 1")
6 liste_1 = [3, 42, 1, -5, 10, 8]
7 print("Liste:", liste_1)
8 print("Sortierte Liste:", insertion_sort(liste_1))
9 print("_"*20)
```





Übung 1



5 2 4 6 1 3

5 2 4 6 1 3

2 5 4 6 1 3

2 4 5 6 1 3

2 4 5 6 1 3

1 2 4 5 6 3

1 2 3 4 5 6

```
Sortiervverfahren.py

1 def insertion_sort(liste):
2     for i in range(1, len(liste)):
3         for j in range(i, 0, -1):
4             if liste[j] < liste[j - 1]:
5                 liste[j], liste[j - 1] = liste[j - 1], liste[j]
6             else:
7                 break
8     return liste
```

Der Algorithmus hat 2 verschachtelte Schleifen mit je Länge (fast) n

=> $O(n^2)$



```
Terminal

Testcase 1
Liste: [3, 42, 1, -5, 10, 8]
Sortierte Liste: [-5, 1, 3, 8, 10, 42]
-----
```

Selectionsort

Selectionsort

klein

groß

2	8	5	3	9	4
---	---	---	---	---	---

0



1



2

3

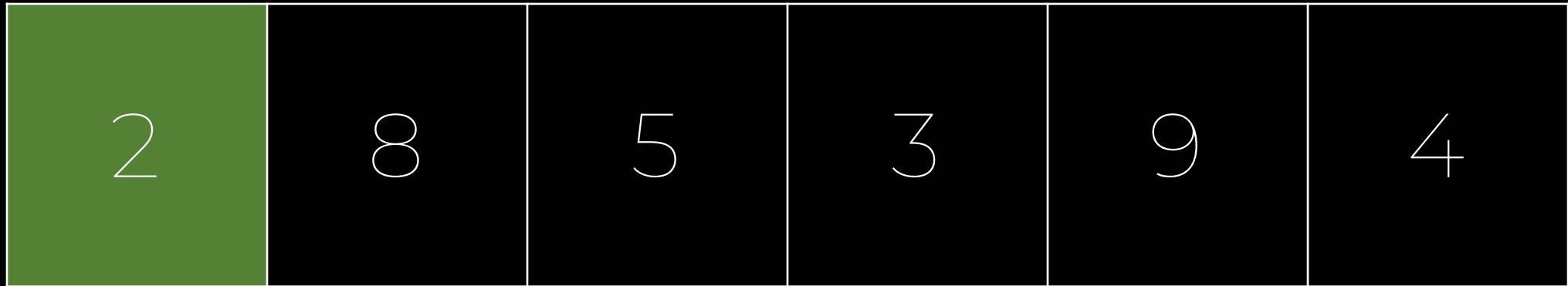
4

5

Selectionsort

klein

groß



0

1

2

3

4

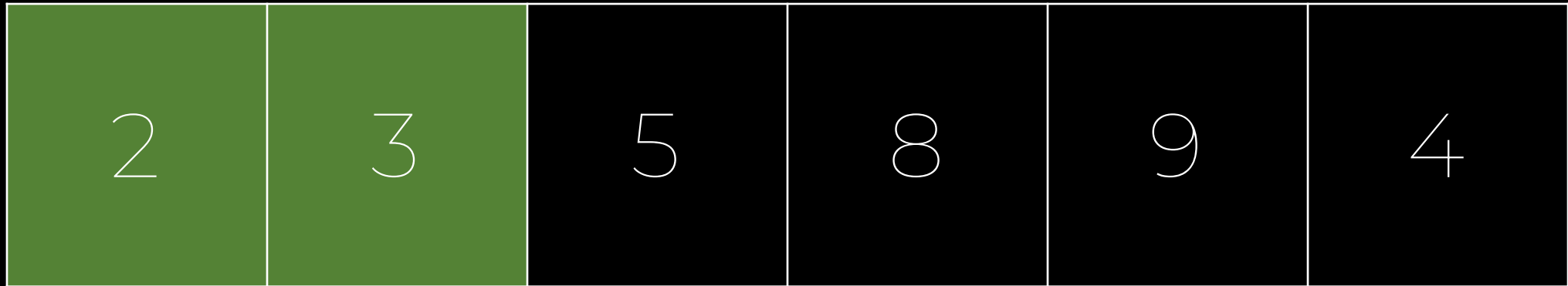
5



Selectionsort

klein

groß



0

1

2

3

4

5



Selectionsort

klein

groß



0

1

2

3

4

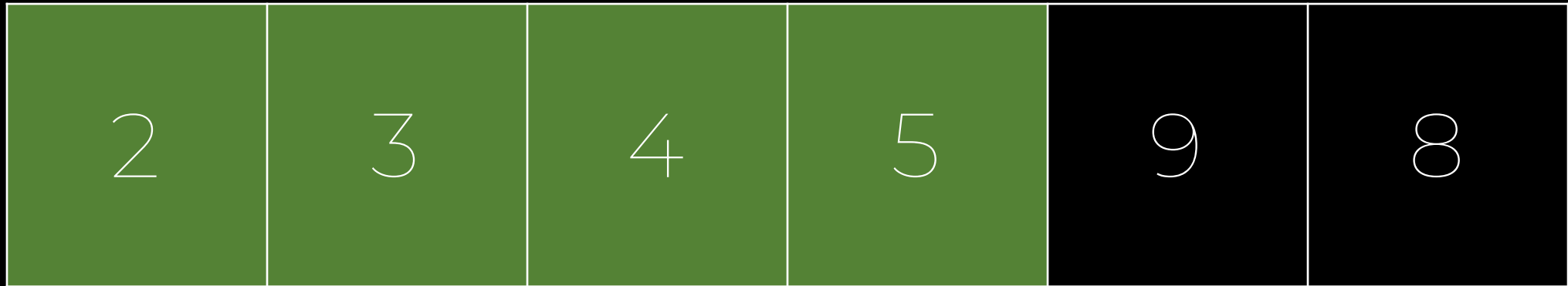
5



Selectionsort

klein

groß



0

1

2

3

4

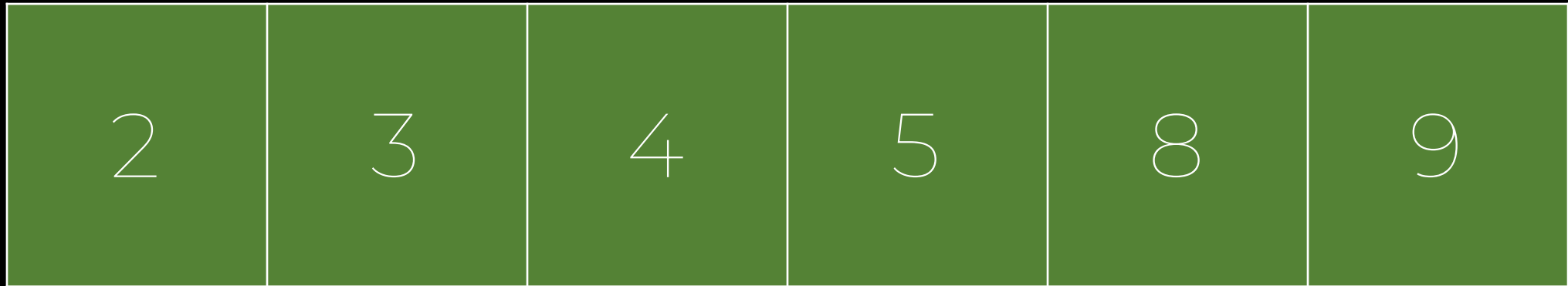
5



Selectionsort

klein

groß



0

1

2

3

4

5



Übung 2

- a) Sortiere folgenden Zahlenfolge mit dem Selectionsort:

5 2 4 6 1 3

- b) Erkläre den Selectionsort in deinen eigenen Worten.
- c) Analysiere die Komplexität (Laufzeit) des Selectionsort.
- d) Zeichne ein Struktogramm zum Selectionsort.
- e) Erweitere dein Programm **Sortierverfahren** um den Selectionsort.

```
Sortierverfahren.py

1 def insertion_sort(liste):
2     ...
3
4 def selection_sort(liste):
5     # DEIN CODE
6     return liste
7
8 print("Testcase 1")
9 liste_1 = [3, 42, 1, -5, 10, 8]
10 print("Liste:", liste_1)
11 print("Insertionsort:", insertion_sort(liste_1))
12 print("Selectionsort:", selection_sort(liste_1))
13 print("_"*20)
```



Übung 2



5 2 4 6 1 3
1 2 4 6 5 3
1 2 4 6 5 3
1 2 3 6 5 4
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6

```
Sortiervverfahren.py

1 def selection_sort(liste):
2     for i in range(len(liste)):
3         min_index = i
4         for j in range(i + 1, len(liste)):
5             if liste[j] < liste[min_index]:
6                 min_index = j
7         liste[i], liste[min_index] = liste[min_index], liste[i]
8     return liste
```

Der Algorithmus hat 2
verschachtelte Schleifen mit
je Länge (fast) n
=> $O(n^2)$



```
Terminal

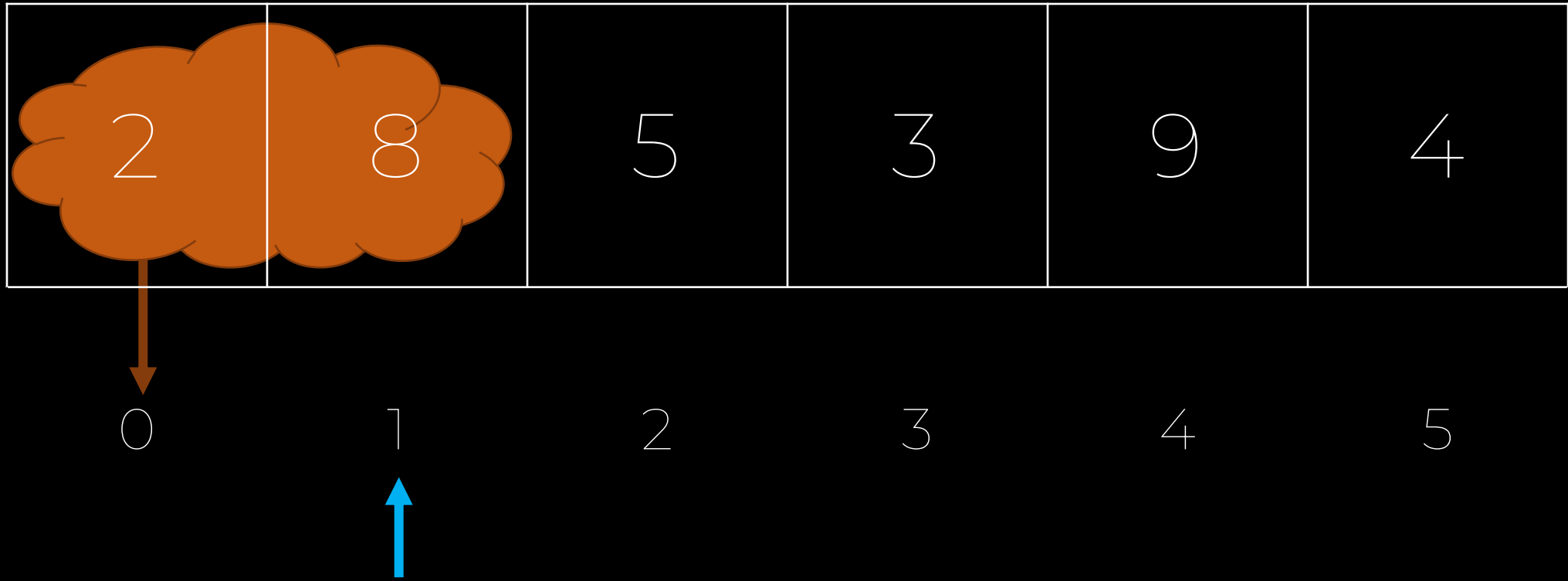
Testcase 1
Liste: [3, 42, 1, -5, 10, 8]
Insertionsort: [-5, 1, 3, 8, 10, 42]
Selectionsort: [-5, 1, 3, 8, 10, 42]
-----
```

Bubblesort

Bubblesort

klein

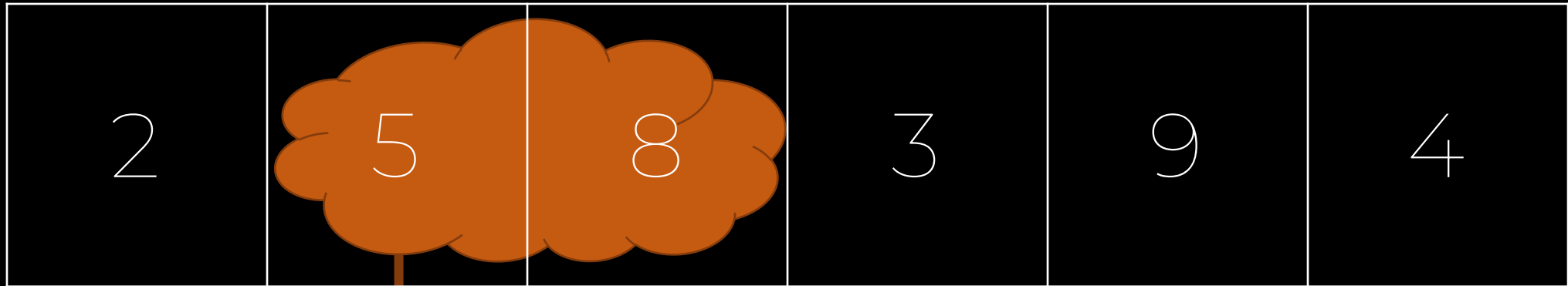
groß



Bubblesort

klein

groß



0

1

2

3

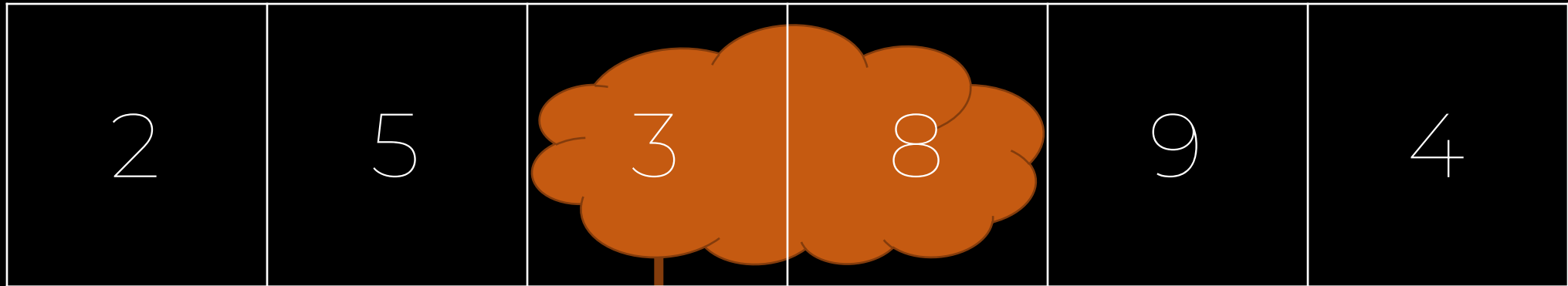
4

5

Bubblesort

klein

groß



0

1

2

3

4

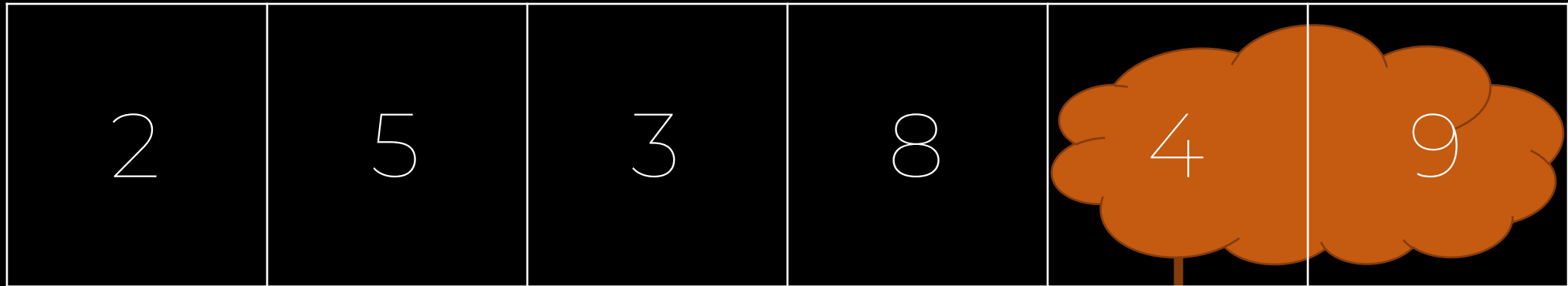
5



Bubblesort

klein

groß



0

1

2

3

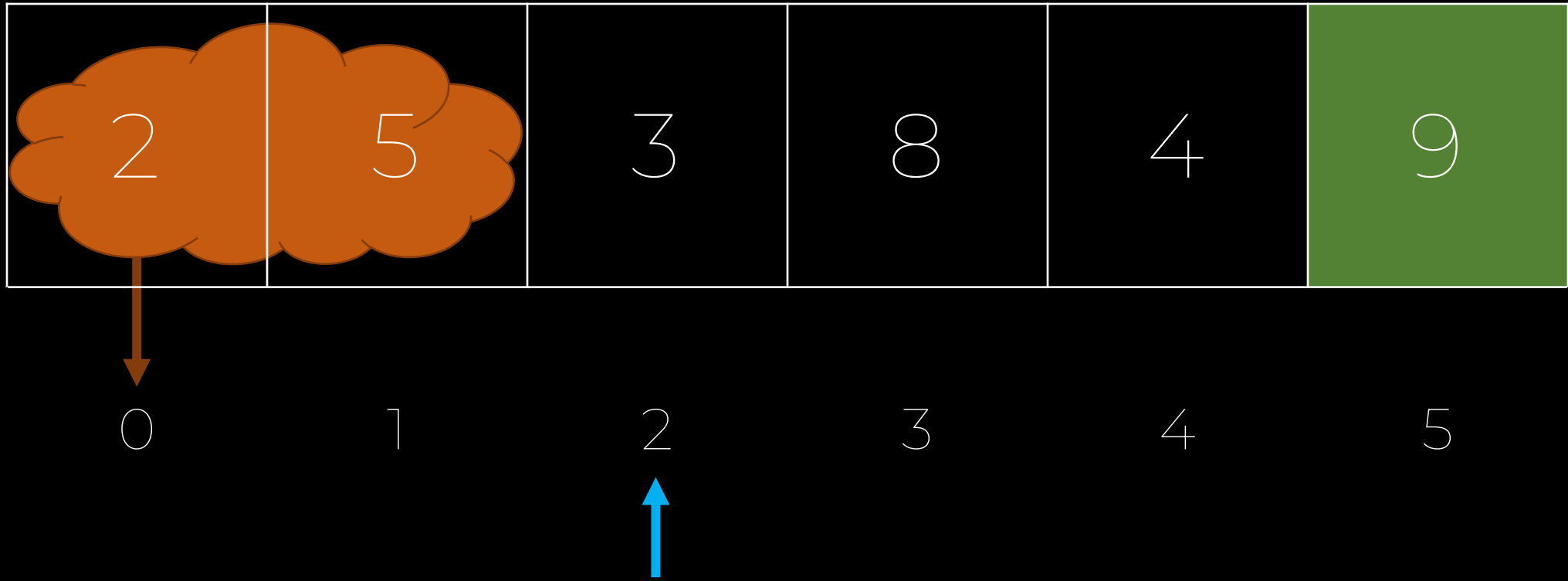
4

5

Bubblesort

klein

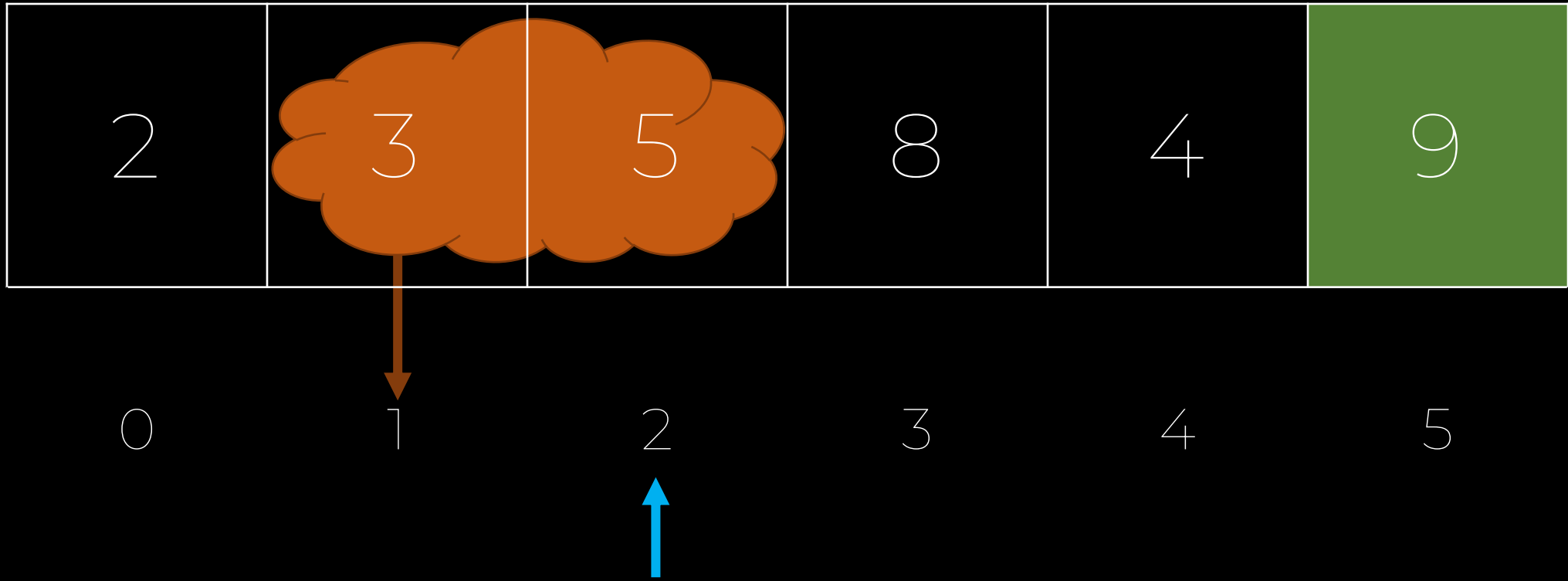
groß



Bubblesort

klein

groß



Bubblesort

klein

groß



0

1

2

3

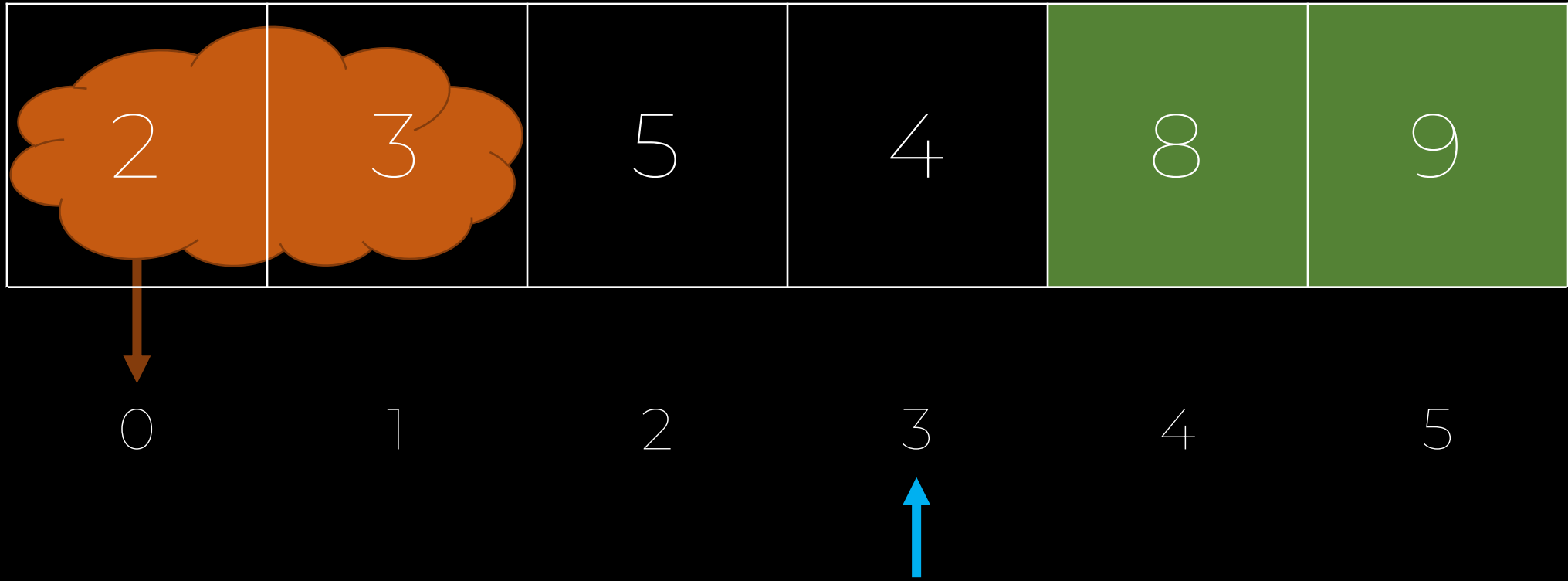
4

5

Bubblesort

klein

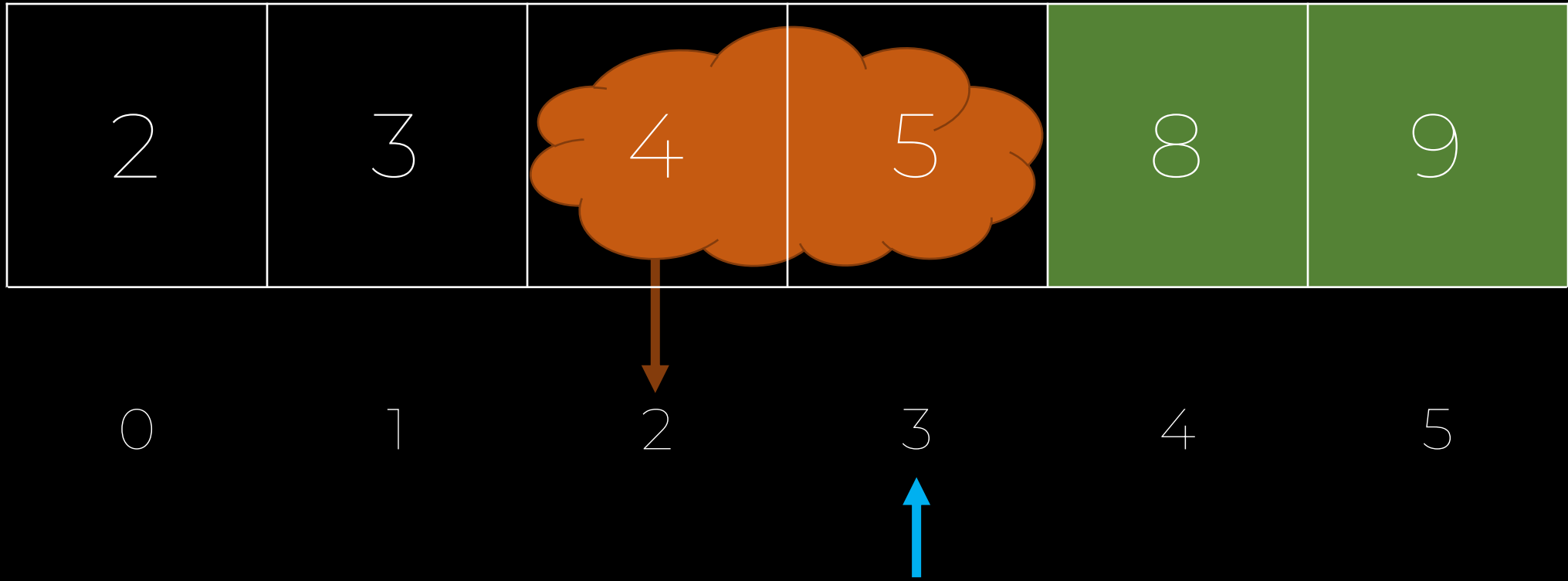
groß



Bubblesort

klein

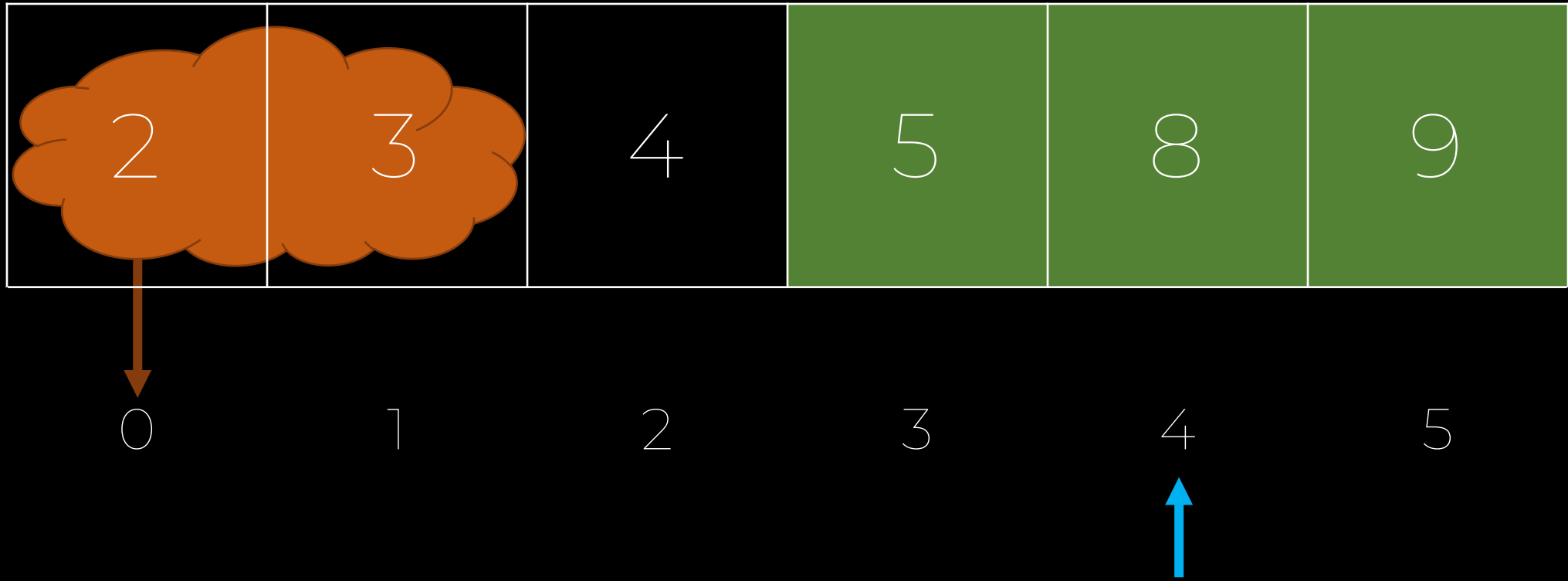
groß



Bubblesort

klein

groß



Bubblesort

klein

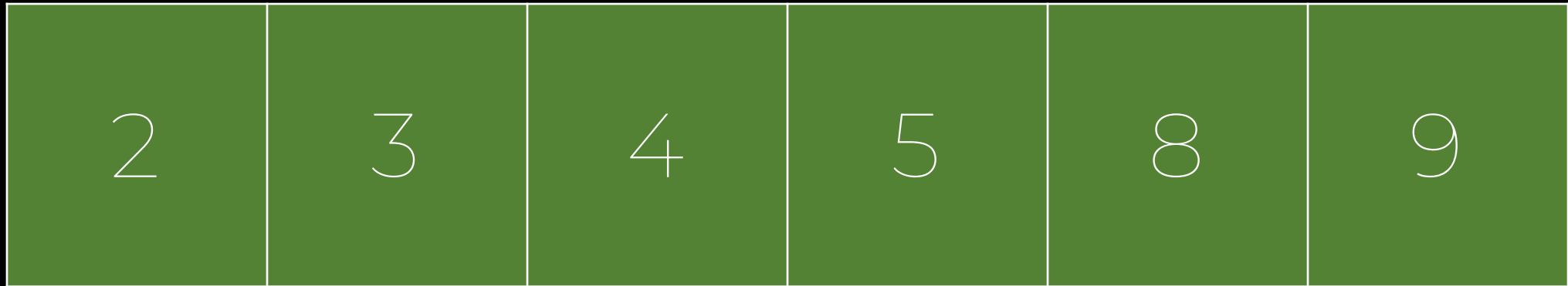
groß



Bubblesort

klein

groß



0

1

2

3

4

5



Übung 3

- a) Sortiere folgenden Zahlenfolge mit dem Bubblesort:

5 2 4 6 1 3

- b) Erkläre den Bubblesort in deinen eigenen Worten.
- c) Analysiere die Komplexität (Laufzeit) des Bubblesort.
- d) Zeichne ein Struktogramm zum Bubblesort.
- e) Erweitere dein Programm **Sortiervverfahren** um den Bubblesort.



Sortiervverfahren.py

```
1 def insertion_sort(liste):
2     ...
3
4 def selection_sort(liste):
5     ...
6
7 def bubble_sort(liste):
8     # DEIN CODE
9     return liste
10
11 print("Testcase 1")
12 liste_1 = [3, 42, 1, -5, 10, 8]
13 print("Liste:", liste_1)
14 print("Insertionsort:", insertion_sort(liste_1))
15 print("Selectionsort:", selection_sort(liste_1))
16 print("Bubblesort:", bubble_sort(liste_1))
17 print("_"*20)
```



Übung 3



5 2 4 6 1 3	2 4 1 3 5 6
2 5 4 6 1 3	2 1 4 3 5 6
2 4 5 6 1 3	2 1 3 4 5 6
2 4 5 6 1 3	1 2 3 4 5 6
2 4 5 1 6 3	1 2 3 4 5 6
2 4 5 1 3 6	1 2 3 4 5 6
2 4 5 1 3 6	1 2 3 4 5 6
2 4 5 1 3 6	
2 4 1 5 3 6	
2 4 1 3 5 6	



Sortiervverfahren.py

```
1 def bubble_sort(liste):
2     for i in range(len(liste)):
3         for j in range(0, len(liste) - i - 1):
4             if liste[j] > liste[j + 1]:
5                 liste[j], liste[j + 1] = liste[j + 1], liste[j]
6     return liste
```

Der Algorithmus hat 2 verschachtelte Schleifen mit je Länge $n \Rightarrow O(n^2)$



Tagebucheintrag

Insertionsort

Selectionsort

Bubblesort



Wochenübung

- a) Recherchiere den Sortieralgorithmus **Mergesort**.
- b) Sortiere nach seinem Schema die Zahlenfolge:
5 2 4 6 1 3
- c) Vergleiche ihn mit den bisherigen Sortverfahren hinsichtlich der Komplexität (Laufzeit).

