

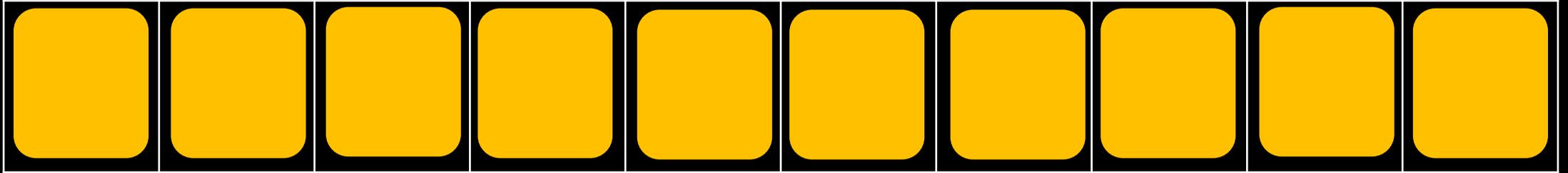
# Informatik Q1 Abels



Was ist eine **Binäre Suche**?

klein

groß



Wo ist die

29

?



klein

groß



$$23 < 29$$

29



... nicht an 4. Stelle ...

klein

groß



$$34 > 29$$

29

... nicht an 7. Stelle ...



klein

groß



$$29 = 29$$

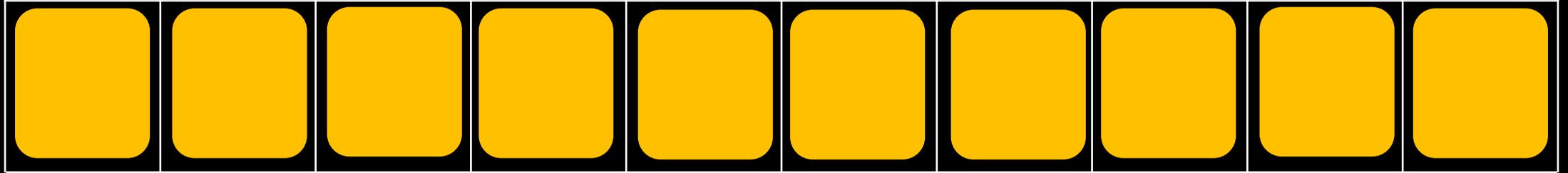
... an 5. Stelle!

29



klein

groß



0

1

2

3

4

5

6

7

8

9



links



rechts

Wo ist die

29

?



klein

groß



0

1

2

3

4

5

6

7

8

9



$$23 < 29$$

mitte

29



... nicht an 4. Stelle ...

klein

groß



0

1

2

3

4

5

6

7

8

9



$34 > 29$



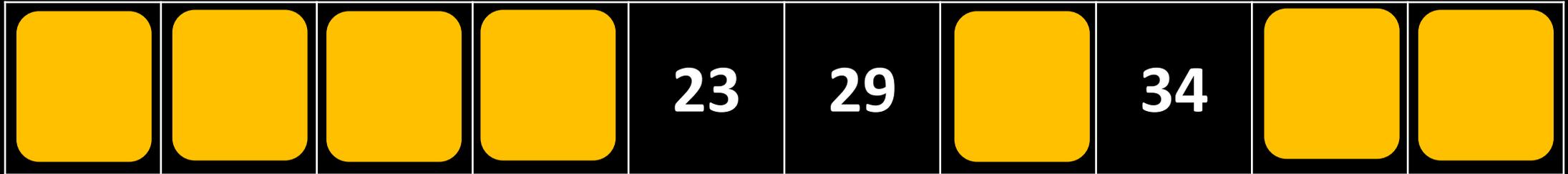
... nicht an 7. Stelle ...

29



klein

groß



0

1

2

3

4

5

6

7

8

9

29 = 29  

... an 5. Stelle!

29





# Übung 1

- a) Formuliere in deinen eigenen Worten einen Algorithmus, ein Element in einer sortierten Liste nach dieser Strategie zu suchen.
- b) Analysiere die Laufzeit des Algorithmus: Wie viele Schritte benötigst du im Worst-Case?
- c) Zeichne zu deinem Algorithmus ein Struktogramm.
- d) Implementiere deinen Algorithmus. Erstelle dazu ein Programm (**Console**) namens **BinarySearch**, in dem das Element **int e** in dem Array **int[] list** gesucht wird. Die Stelle, an der das Element gefunden wurde, soll in der Konsole ausgegeben werden.
- e) Erweitere dein Programm, sodass auch der Fall berücksichtigt wird, dass das Element nicht im Array vorhanden ist.



# Tagebucheintrag

Binäre Suche



# Wochenübung

Erweitere dein Programm **SuchenUndSortieren** um die Funktion

```
int binarySearch(int e, int[] list)
```

, die ein Element und einen Array übergeben bekommt und den Index des gefundenen Elements oder -1 zurückgibt. In der **main**-Methode sollen alle nötigen Testcases (2!) getestet werden.