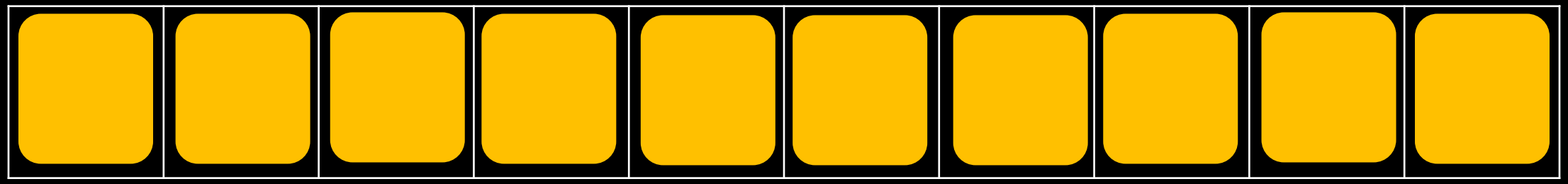
Informatik Q1 Abels

Binäre Suche

Binäre Suche

klein groß



Wo ist die 29 ?

Binäre Suche

klein groß

$23 < 29$

29

... nicht an 4. Stelle ...

Binäre Suche

klein

groß

34 > 29

29

... nicht an 7. Stelle ...

Binäre Suche

klein

groß

				23	29		34		

$29 = 29$

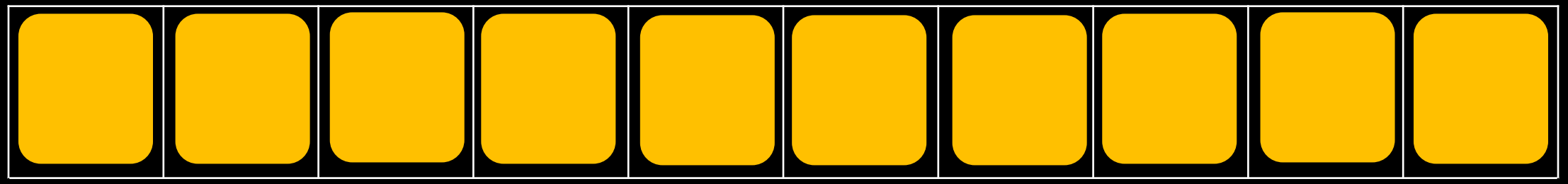
29

... an 5. Stelle!

Binäre Suche

klein

groß



0 1 2 3 4 5 6 7 8 9

links

rechts

Wo ist die 29 ?

Binäre Suche

klein

groß

0 1 2 3 4 5 6 7 8 9

mitte

$23 < 29$

29

... nicht an 4. Stelle ...

Binäre Suche

klein groß

0 1 2 3 4 5 6 7 8 9

34 > 29

... nicht an 7. Stelle ...

29

Binäre Suche

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) Implementiere deinen Algorithmus. Erstelle dazu ein Programm namens **`BinarySearch.java`**, 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.

d) Erweitere dein Programm, sodass auch der Fall berücksichtigt wird, dass das Element nicht im Array vorhanden ist.

BinarySearch.java

```java
public class BinarySearch {
    public static void main(String[] args) {
        int[] list = {-4, 1, 3, 9, 10, 42};
        int e = 42;
        int index, left, middle, right;

        // DEIN CODE

        System.out.println(index);
    }
}
```

Terminal

```
5
```

Übung 1

Im Worst-Case halbiert der Algorithmus die Liste so oft, bis nur noch 1 Element übrig ist => O(log n)

BinarySearch.java

```java
public class BinarySearch {
    public static void main(String[] args) {
        int[] list = {-4, 1, 3, 9, 10, 42};
        int e = 42;
        int index = -1;

        int left = 0;
        int right = list.length - 1;
        while (left <= right) {
            int middle = left + (right - left) / 2;
            if (list[middle] == e)
                index = middle;
            if (list[middle] < e)
                left = middle + 1;
            else
                right = middle - 1;
        }

        System.out.println(index);
    }
}
```

Tagebucheintrag

Binäre Suche

Wochenübung

Erweitere dein Programm **SuchenUndSortieren.java** um die Funktion

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

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

SuchenUndSortieren.java

```java
public class SuchenUndSortieren {
    public static void main(String[] args) {
        // DEIN CODE
    }

    public static int binarySearch(int e, int[] list) {
        // DEIN CODE
    }
}
```