DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Tutorium Programmierung 2

28.04.2025

Janens Kurzke und Fabian Bauriedl

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Inhalt

1. Organisatorisches
2. Rekursion
3. Suchalgorithmen
4. Sortieralgorithmen

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Organisatorisches

> Steffen und Mario behandeln die Themen in unterschiedlicher Reihenfolge

Organisatorisches

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Komm in die Gruppe!

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Rekursion

```
factorial( n ):
if n == 1:
    return 1
else:
    return n * factorial(n-1):
```

factorial(n) =

www.penjee.com

Suchalgorithmen

Linear Search

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Suchalgorithmen

> Linear Search

```
public static int search(int array[], int N, int x) {
    for (int i = 0; i < N; i++) {
        if (array[i] == x)
            return i;
    }
    return -1;
}
```

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Suchalgorithmen

Binäre Suche

Search for 47

0	4	7	10	14	23	45	47	53

Suchalgorithmen

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Binäre Suche

```
static int binarySearch(int array[], int l, int r, int x){
    while (l <= r) {
        int m = (l + r) / 2;

        if (array[m] == x) {
            return m;
        } else if (array[m] > x) {
            r = m - 1;
        } else {
            l = m + 1;
        }
    }
    return -1;
}
```

Suchalgorithmen

Interpolationssuche

Suchalgorithmen

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

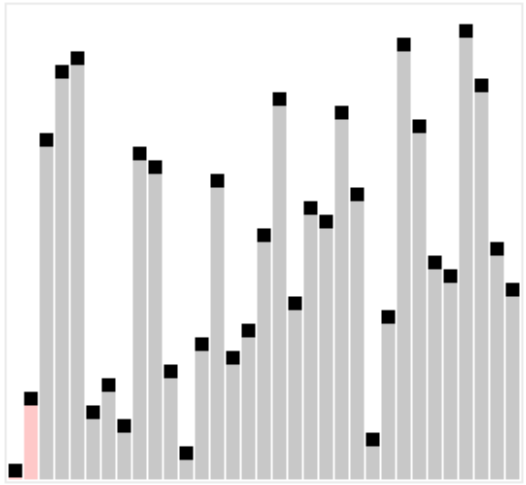
Interpolationssuche

```
public static int interpolationSearch(int array[], int lo, int hi, int x) {
    int pos;

    if (lo <= hi && x >= array[lo] && x <= array[hi]) {
        pos = lo + (((hi - lo) / (array[hi] - array[lo])) * (x - array[lo]));

        if (array[pos] == x) return pos;

        if (array[pos] < x) return interpolationSearch(array, pos + 1, hi, x);

        if (array[pos] > x) return interpolationSearch(array, lo, pos - 1, x);
    }
    return -1;
}
```

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Sortieralgorithmen

Selection Sort



Sortieralgorithmen

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

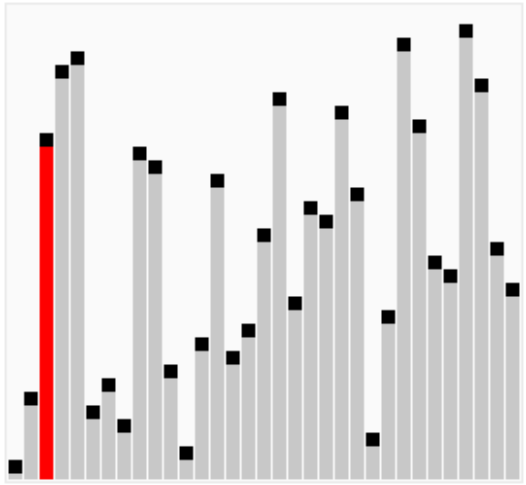
Selection Sort

```
public static void selectionSort(int[] array) {
    int arrayLength = array.length;

    for (int currentPosition = 0; currentPosition < arrayLength - 1; currentPosition++) {
        int minimumIndex = currentPosition;

        for (int index = currentPosition + 1; index < arrayLength; index++) {
            if (array[index] < array[minimumIndex]) {
                minimumIndex = index;
            }
        }

        int temporary = array[minimumIndex];
        array[minimumIndex] = array[currentPosition];
        array[currentPosition] = temporary;
    }
}
```

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Sortieralgorithmen

Bubble Sort



Sortieralgorithmen

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

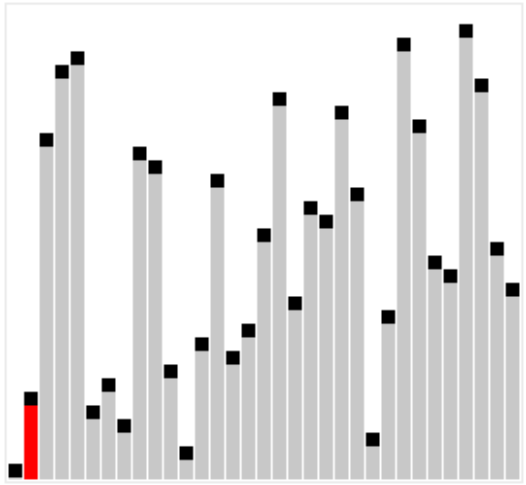
Bubble Sort

```
public static void bubbleSort(int[] array) {
    int arrayLength = array.length;

    for (int pass = 0; pass < arrayLength - 1; pass++) {
        for (int index = 0; index < arrayLength - pass - 1; index++) {
            if (array[index] > array[index + 1]) {
                int temporary = array[index];
                array[index] = array[index + 1];
                array[index + 1] = temporary;
            }
        }
    }
}
```

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Sortieralgorithmen

Insertion Sort



Sortieralgorithmen

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

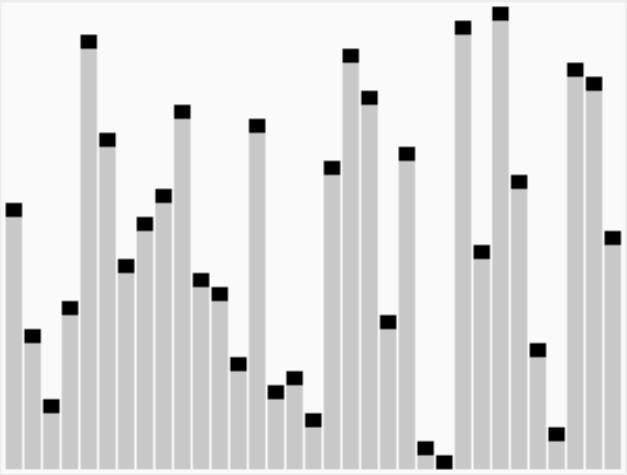
Insertion Sort

```
public static void insertionSort(int[] array) {
    int arrayLength = array.length;

    for (int index = 1; index < arrayLength; index++) {
        int currentValue = array[index];

        int position = index;
        while (position > 0 && array[position - 1] > currentValue) {
            array[position] = array[position - 1];
            position--;
        }
        array[position] = currentValue;
    }
}
```

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Sortieralgorithmen

Quick Sort



DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Sortieralgorithmen

> Quick Sort

```
public static void quickSort(int[] array, int left, int right) {
    if (left < right) {
        int partitionIndex = partition(array, left, right);
        quickSort(array, left, partitionIndex - 1);

        quickSort(array, partitionIndex + 1, right);
    }
}
```

Sortieralgorithmen

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Quick Sort

```
private static int partition(int[] array, int left, int right) {
    int pivot = array[right];
    int i = left - 1;
    for (int j = left; j < right; j++) {
        if (array[j] < pivot) {
            i++;
            int temporary = array[i];
            array[i] = array[j];
            array[j] = temporary;
        }
    }
    int temporary = array[i + 1];
    array[i + 1] = array[right];
    array[right] = temporary;
    return i + 1;
}
```

Sortieralgorithmen

Merge Sort

Sortieralgorithmen

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

> Merge Sort

```
public static void mergeSort(int[] array, int left, int right) {
    if (left < right) {
        int middle = (left + right) / 2;
        mergeSort(array, left, middle);

        mergeSort(array, middle + 1, right);
        merge(array, left, middle, right);
    }
}
```

Sortieralgorithmen

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Merge Sort

```
private static void merge(int[] array, int left, int middle, int right) {
    int leftArrayLength = middle - left + 1;
    int rightArrayLength = right - middle;

    int[] leftArray = new int[leftArrayLength];
    int[] rightArray = new int[rightArrayLength];

    for (int i = 0; i < leftArrayLength; ++i)
        leftArray[i] = array[left + i];
    for (int j = 0; j < rightArrayLength; ++j)
        rightArray[j] = array[middle + 1 + j];

    int i = 0;
    int j = 0;
    int k = left;

    while (i < leftArrayLength && j < rightArrayLength) {
        if (leftArray[i] <= rightArray[j]) {
            array[k] = leftArray[i];
            i++;
        } else {
            array[k] = rightArray[j];
            j++;
        }
        k++;
    }
    while (i < leftArrayLength) {
        array[k] = leftArray[i];
        i++;
        k++;
    }
    while (j < rightArrayLength) {
        array[k] = rightArray[j];
        j++;
        k++;
```

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Zeit zum Üben