NORDAKADEMIE
HOCHSCHULE DER WIRTSCHAFT

THEORIE PRAXIS

Algorithmen und Datenstrukturen

Teil 5: Aufgaben

Studiengang Wirtschaftsinformatik

Kai Hufenbach

Aufgabe 1: Wiederholung Komplexitätsbestimmung I

Bestimmen Sie die Laufzeitkomplexität von:

1)

```
int[] array = createArrayOfSize(n);

long sum = 0;

long time = System.nanoTime();
for (int i = 0; i < n; i++) {
  sum += array[i];
}
```

2)

```
private static void insertionSort(int[] elements) {
  for (int i = 1; i < elements.length; i++) {
    int elementToSort = elements[i];
    int j = i;
    while (j > 0 && elementToSort < elements[j - 1]) {
      elements[j] = elements[j - 1];
      j--;
    }
    elements[j] = elementToSort;
  }
}
```

Bestimmen Sie die Laufzeitkomplexität von:

1)

```
static int power (int k, int n) {
   if (n == 0)
      return 1;
   else
      return k * power (k, n - 1);
}
```

2)

Anfang: fib(0) = 1 fib(1) = 1
Schritt: fib(n + 1) = fib(n) + fib(n - 1)

3) Wie könnte mittels Potenzgesetze eine effizientere Implementation von `power` aussehen?

Aufgabe 3: Divide-and-Conquer

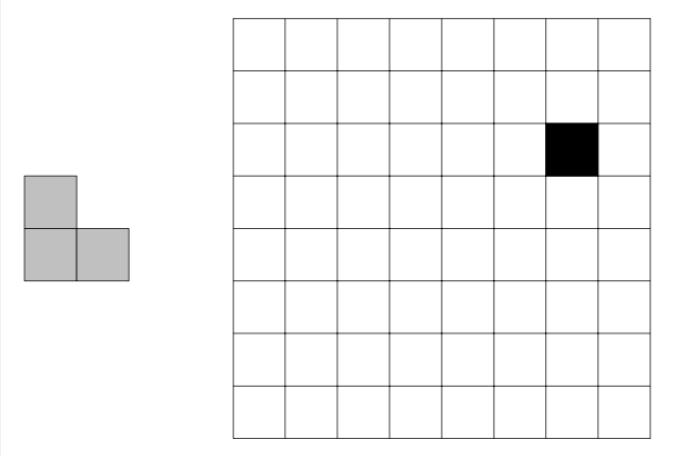
Erstellen Sie einen Algorithmus im Pseudocode, der nach Divide-and-Conquer das größte Element in einem Array findet.

- Was passiert, wenn es mehrere Elemente gibt, die das größte Element repräsentieren?
- Ermitteln Sie die Komplexität des Algorithmus
- Wie verhält sich der Algorithmus im Vergleich zu einem Brute-Force-Ansatz?

Aufgabe 4: Tromino

Ein Tromino ist ein Baustein in L-Form, der aus drei zusammenhängenden Quadraten besteht. Das Problem besteht darin, ein beliebiges $2^n \; x \; 2^n$ großes Schachbrett so zu füllen, dass genau ein Feld leer bleibt.

Dabei darf sich kein Tromino überlappen. Der Baustein darf jedoch, ähnlich bei Tetris, gedreht werden.



Gestalten Sie strukturell eine Divide-and-Conquer Lösung für dieses Problem.

Aufgabe 5: Negatives vor Positivem

Gestalten Sie einen Algorithmus, der ein beliebiges Array aus n Zahlen so anordnet, dass alle negativen Zahlen vor den positiven Zahlen stehen.

Z.B. könnte die Lösung für:

```
[5, 9, -2, -5, 7, 1]
```

Folgendermaßen aussehen:

```
[-2, -5, 5, 9, 7, 1]
```

Aufgabe 6: Holländische Flagge (erweitert Aufgabe 5)

Gestalten Sie einen Algorithmus, der ein beliebiges Array aus den Farben Rot, Weiß, Blau so anordnet, dass alle R‘s vor den W‘s und alle W‘s vor den B‘s stehen

Z.B. könnte die Lösung für:

```
[W, R, R, R, B, W, B]
```

Folgendermaßen aussehen:

```
[R, R, R, W, W, B, B]
```