ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Dr. Jochen Hoenicke
Alexander Nutz

21.7.2014
Abgabe: Montag 28.7.2014
via Daphne/Subversion

12. Übungsblatt zur Vorlesung
Einführung in die Informatik

Aufgabe 1: Laufzeit von Bubblesort
Betrachten Sie untenstehendes Programm zum Sortieren eines Arrays.

(a) Versehen Sie jedes Statement und jeden Block (also zum Beispiel jede for-Schleife und die Methode), mit einer Worst-Case-Laufzeitangabe in $\mathcal{O}$-Notation im Stil der Vorlesung. Benutzen Sie als Eingabegröße die Länge des Arrays a.

(b) Was ist die Best-Case Laufzeit dieser Methode? Geben Sie jeweils eine Eingabe an, die zur Best-Case-Laufzeit und eine die zur Worst-Case-Laufzeit führt.

```
void bubbleSort(int[] a) {
  boolean swapped = false;
  int j = a.length;
  do {
    swapped = false;
    for (i = 0; i < j - 1; i++) {
      if (A[i] > A[i+1]) {
        int tmp = a[j];
        a[j] = a[i];
        a[i] = tmp;
        swapped = true;
      }
    }
    j--;
  } while (swapped);
}
```

Aufgabe 2: $\mathcal{O}(\log(n))$
In der Vorlesung wurde behauptet, dass gilt

$$\mathcal{O}(\log_2(n)) = \mathcal{O}(\log_c(n))$$

für beliebige Konstanten $c \in \mathbb{N}$. Begründen Sie diese Aussage.

Aufgabe 3: Laufzeit

Was ist die Worst-Case-Laufzeit des folgenden Programms? Schreiben Sie wieder neben jedes Statement/jeden Block eine Komplexitätsangabe in $\mathcal{O}$-Notation. Wo die Java Standardbibiothek aufgerufen wird (also bei HashSet.add(..) und Object.clone()), dürfen Sie Konstantzeit annehmen.

```
HashSet<HashSet<T>> computePowerSet(HashSet<T> inputSet) {
  HashSet<HashSet<T>> powerSet = new HashSet<>();
  for (T item : inputSet) {
    HashSet<HashSet<T>> setsToAdd = new HashSet<>();
    if (powerSet.isEmpty()) {
      setsToAdd.add(new HashSet<T>());
      HashSet<T> newSet = new HashSet<>();
      newSet.add(item);
      setsToAdd.add(newSet);
    } else {
      for (HashSet<T> subset : powerSet) {
        setsToAdd.add((HashSet<T>)subset.clone());
        subset.add(item);
      }
    }
    for (HashSet<T> set : setsToAdd) {
      powerSet.add(set);
    }
  }
  return powerSet;
}
```