

# Theoretische Informatik 1 / Algorithmentheorie

Wintersemester 2012/13

Prof. Dr. Georg Schnitger,  
Dipl. Inf. Bert Besser,  
Dipl. Inf. Matthias Poloczek

Arbeitsgruppe Theoretische Informatik, Institut für Informatik

---



## Blatt 1

Ausgabe: 18.10.2012  
Abgabe: 25.10.2012 **vor** der Vorlesung

Bitte besuche die Homepage der Veranstaltung unter

<http://www.thi.informatik.uni-frankfurt.de/TheoretischeInformatik/1/WS1213/>.

Dort werden neben dem Skript zur Vorlesung und den Übungsaufgaben weitere interessante Links sowie aktuelle Information bereit gestellt.

Generell gilt für alle Übungsaufgaben (sofern nicht explizit anders angegeben):

- Erläutere Deinen Lösungsansatz.
- Beweise Deine Aussagen bzw. gib Deinen Lösungsweg an.
- Strukturiere Deine Gedanken, bevor Du sie niederschreibst.

Teamarbeit ist ausdrücklich erlaubt, aber alle Lösungen müssen selbständig aufgeschrieben werden. Für *jeden* Betroffenen von plagierten Lösungen gilt: beim ersten Mal wird das entsprechende Blatt mit null Punkten bewertet; im Wiederholungsfall werden keine Bonuspunkte für die Klausur angerechnet.

- Schreibe auf Deine Abgabe stets
  - a) Deinen Namen,
  - b) Deine Matrikelnummer sowie
  - c) den Übungstermin.

Bitte hefte mehrseitige Abgaben zusammen.

Beachte, dass der Sortieralgorithmus Heapsort  $n$  Zahlen mit der worst-Case Laufzeit  $\Theta(n \cdot \log(n))$  sortiert.

### 1.1. Aufgabe (4+4)

*Elementare Sortierverfahren*

**Bestimme** für die Sortieralgorithmen *Insertion Sort*, *Bubble Sort* und *Selection Sort* aus dem Vorlesungsskript asymptotisch exakt (d.h. unter Verwendung der  $\Theta$ -Notation) die Anzahl der Vergleiche und die Anzahl der Vertauschungen auf den folgenden Zahlenfolgen. Eine Begründung ist wie immer erforderlich.

- a)  $N, 1, 2, 3, \dots, N - 1$
- b)  $1, \frac{N}{2} + 1, 2, \frac{N}{2} + 2, \dots, \frac{N}{2}, N$  ( $N$  gerade)

### 1.2. Aufgabe (8)

*Herbst*

An einem Herbsttag möchte die Stadt das Laub auflesen. Am Morgen sind die Laubvolumen  $v(s_i)$  der Stadtteile  $S = \{s_1, \dots, s_n\}$  bekannt. Außerdem sind verfügbare Transporter  $T = \{t_1, \dots, t_n\}$  zusammen mit ihren Kapazitäten  $c(t_i)$  gegeben. Die folgenden Einschränkungen gelten:

- Jeder Stadtteil wird von genau einem Transporter bedient oder gar nicht.
- Jeder Transporter kann maximal einen Stadtteil vom Laub befreien.

Gib einen möglichst schnellen Algorithmus an, der eine Zuordnung von Transporter zu Stadtteil ausgibt, so dass die Anzahl der gereinigten Stadtteile maximiert wird.

### 1.3. Aufgabe (8)

*Konservative Sortieralgorithmen*

Ein Sortieralgorithmus  $A$  heißt *konservativ*, wenn in jedem Schritt nur die Inhalte *benachbarter* Zellen des Eingabe-Arrays vertauscht werden. Beachte, dass Bubble-Sort ein konservativer Sortieralgorithmus ist. Auch Selection Sort und Insertion Sort lassen sich als konservative Sortieralgorithmen auffassen. Zeige, dass *jeder* konservative Sortieralgorithmus langsam sein *muss*. Genauer: Zeige, dass  $A$ 's worst-case Laufzeit  $\Omega(n^2)$  beträgt, wenn  $n$  Zahlen zu sortieren sind.