Algorithmen und Berechnungskomplexität I, WS 12/13
Aufgabenblatt 2
Universität Bonn, Institut für Informatik, Abteilung I

- *Die Lösungen können bis Dienstag, 30.10., 12:15 Uhr in den Postkasten im AVZ III eingeworfen werden (vom Haupteingang im kleinen Raum auf der linken Seite). Gebt bitte immer gut sichtbar auf dem Deckblatt die Gruppennummer (A-I) an, wie auf der Vorlesungswebseite angegeben.*
- *Hier kann man sich zu unserer Mailingliste anmelden: https://lists.iai.uni-bonn.de/mailman/listinfo.cgi/vl-algber1*
- *Wer noch keiner Übungsgruppe zugeordnet ist und dennoch am Übungsbetrieb teilnehmen möchte, kontaktiert bitte Rainer Penninger (penninge@cs.uni-bonn.de).*

**Aufgabe 5: Schachbrett (4 Punkte)**

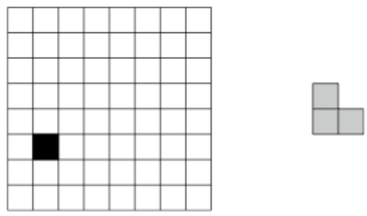

Abbildung 1: Schachbrett und L-förmiger Stein.

Beweisen Sie für alle natürlichen Zahlen $n \geq 1$ per Induktion: Jedes $2^n \times 2^n$-Schachbrett, aus dem man ein beliebiges Feld herausgenommen hat, kann man ohne Lücken oder Überlappungen mit L-förmigen Steinen überdecken, die jeweils drei Felder bedecken (also mit um ein Vielfaches von 90° gedrehten Kopien des Steines in Abbildung 1 oben rechts).

*Bitte wenden!*

## Aufgabe 6: Laufzeitbestimmung Dichtestes Punktepaar

Analysieren Sie die Gesamtlaufzeit des *Divide and Conquer*-Algorithmus, wie er in der Vorlesung vorgestellt wurde, zum Bestimmen eines dichtesten Punktepaares einer Menge von Punkten in der Ebene. Stellen Sie hierzu eine Rekursionsgleichung auf, die Sie lösen, um die Laufzeit des Verfahrens in $\Theta$-Notation anzugeben. Sie können die vereinfachenden Annahmen aus der Vorlesung verwenden und folglich als Rekursionsgleichung Gleichung (2.4) im Vorlesungsskript zugrunde legen.

## Aufgabe 7: Entwurf und Analyse eines Algorithmus

Gegeben seien $n$ Punkte in der Ebene. Geben Sie einen Divide-and-Conquer-Algorithmus an, der die Ebene so lange in Flächen unterteilt, bis in jeder Unterteilungsfläche nur noch *genau* ein Punkt liegt. Eine vorhandene Region darf hierbei durch eine Gerade (die endet, sobald sie eine Regionsgrenze schneidet) in zwei Teile aufgeteilt werden. Dabei soll die Unterteilungshierarchie in einem Baum abgelegt werden, siehe Abbildung 2. Nehmen Sie dabei an, dass Prozeduren für grundlegende (geometrische) Operationen existieren, wie bspw. eine Funktion zum Überprüfen, ob eine Fläche nur noch genau einen Punkt enthält. Wie groß ist die Tiefe des Baumes im Worst-Case?

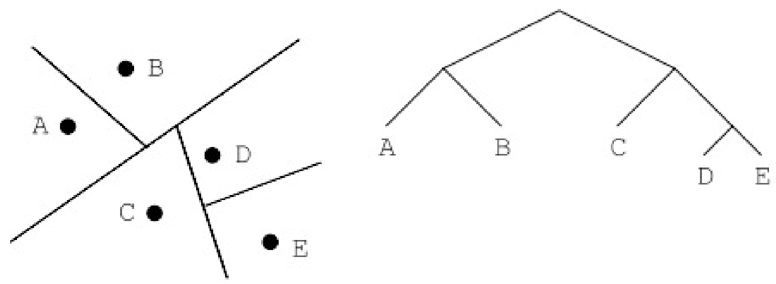


Abbildung 2: Beispiel zu Aufgabe 3.

**Aufgabe 8:** **Einen Algorithmus analysieren**

Betrachten Sie den folgenden rekursiven Algorithmus, der auf einem Array mit natürlichen Zahlen operiert:

**Algorithmus** *QS(array A)*
QSrek(A, 1, size(A));

**Algorithmus** *QSrek(Array A, integer l, r)*
if $l < r$ then
    $p :=$ part$(A, l, r)$;
    QSrek(A, l, p-1);
    QSrek(A, p+1, r);

**Algorithmus** *part(A, l, r)*
$x := A[r]$;
$i := l - 1$;
for $j = l$ to $r - 1$ do
    if $A[j] \leq x$ then
        $i := i + 1$;
        Vertausche $A[i]$ und $A[j]$;
Vertausche $A[i + 1]$ mit $A[r]$
return i+1;

Vollziehen Sie die Funktionsweise des Algorithmus anhand eines Beispielarrays mit mindestens 10 Einträgen nach. Was macht dieser Algorithmus?