Arbeitsblatt: Festigung der Vorstellung einer Rekurrenzgleichung

1.Teil: Vollständige Induktion

Problem 1: Anzahl Diagonalen im n-Eck

Auftrag 1.1: Zeichne je ein Viereck, ein Fünfeck und ein Sechseck und zähle jeweils die Diagonalen.

Auftrag 1.2: Suche eine Regel bzw. eine Formel für die Anzahl der Diagonalen in einem n-Eck. Prüfe deine Regel an einem Beispiel.

Auftrag 1.3: Stelle nun eine Rekurrenzgleichung für die Anzahl der Diagonalen im n-Eck auf.

Auftrag 1.4: Verifiziere, dass die Formel aus Auftrag 1.2 die Rekurrenzgleichung erfüllt.

Auftrag 1.5: In Auftrag 1.2 wurde eine Formel für die Anzahl der Diagonalen im n-Eck gefunden, dazu wurden kombinatorische Überlegungen verwendet, alternativ kann eine solche explizite Gleichung für T(n) direkt aus der Rekurrenzgleichung gewonnen werden: Entwickle diese explizite Gleichung durch wiederholtes Einsetzen.

Problem 2: Anzahl Verbindungsstrecken zwischen n Punkten in allgemeiner Lage

Auftrag 2.1: Zähle alle möglichen Verbindungsstrecken, wenn vier bzw. sechs Punkte in allgemeiner Lage gegeben sind.

Auftrag 2.2: Suche eine Formel bzw. eine Regel für die Anzahl Verbindungsstrecken von n Punkten in allgemeiner Lage.

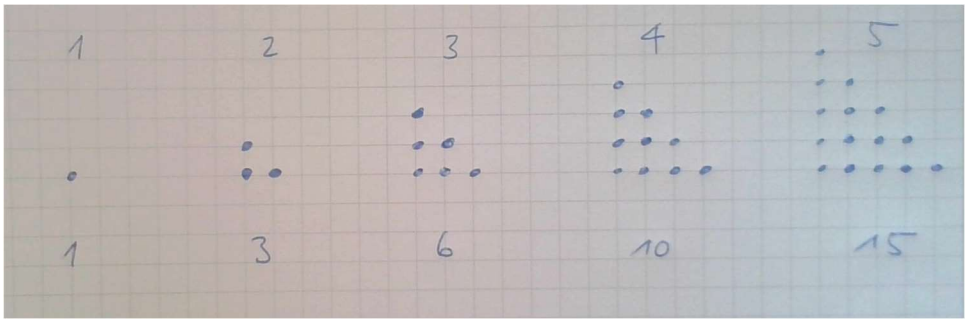
Auftrag 2.3: Stelle nun eine Rekurrenzgleichung für die Anzahl der Verbindungsstrecken von n Punkten auf.

Auftrag 2.4: Verifiziere, dass die Formel aus Auftrag 2.2 die Rekurrenzgleichung erfüllt.

Auftrag 2.5: Entwickle die explizite Gleichung von T(n) durch wiederholtes Einsetzen in die Rekurrenzgleichung.
(Alternative zu den kombinatorischen Überlegungen in Auftrag 2.2).

Problem 3: Anzahl Punkte im Punktmuster der Dreieckszahlen

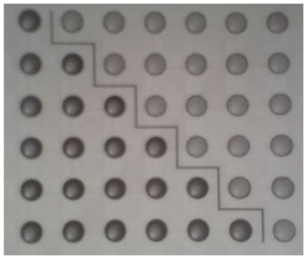
Wir betrachten die Folge der Dreieckszahlen:



Auftrag 3.1: Stelle eine Formel für die Anzahl Punkte der n.ten Dreieckszahl auf.

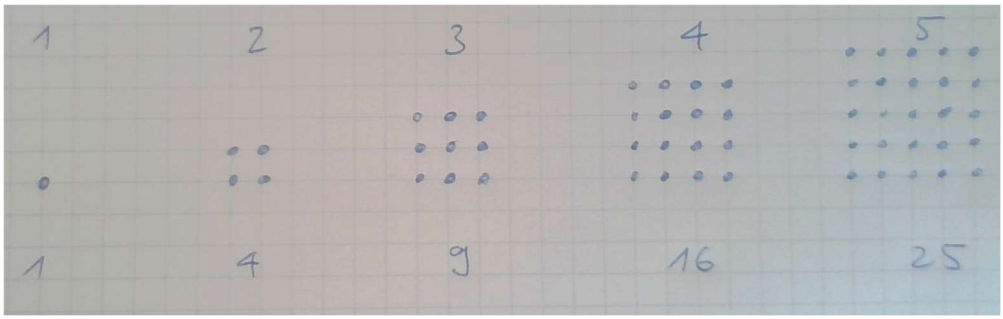
Auftrag 3.2: Stelle eine Rekurrenzgleichung für die Anzahl Punkte der n.ten Dreieckszahl auf.

Auftrag 3.3: Zu Prüfen, ob die Formel aus dem Auftrag 3.1 die Rekurrenzgleichung erfüllt, wollen wir uns sparen, weil die Umformungen ganz analog wie oben im Auftrag 2.4 sind. Hingegen soll hier noch ein eindrücklicher Beweis betrachtet werden. Das folgende Bild „beweist“ - ohne Worte - eine Summenformel, welche?



Problem 4: Anzahl Punkte im Punktmuster der Quadratzahlen

Ähnlich können wir eine Folge von Quadratzahlen betrachten:



Auftrag 4.1: Die Formel für die n.te Quadratzahl (= Anzahl Punkte der n.ten Quadratfigur) ist offensichtlich n^2. Weniger offensichtlich ist die dazugehörige Rekurrenzgleichung, wie lautet sie?

Auftrag 4.2: Verifiziere, dass die Formel T(n) = n^2 die Rekurrenzgleichung erfüllt.

Auftrag 4.3: Entwickle die explizite Gleichung von T(n) durch wiederholtes Einsetzen in die Rekurrenzgleichung.
(Dies sollte die offensichtlich korrekte Formel T(n) =n^2 bestätigen.)

2.Teil: Effiziente Algorithmen für die Matrixmultiplikation

Problem 5: Matrixmultiplikation

Eingabe: Zwei $(n \times n)$-Matrizen A=$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}$ und B=$\begin{pmatrix} b_{11} & \dots & b_{1n} \\ \vdots & & \vdots \\ b_{n1} & \dots & b_{nn} \end{pmatrix}$

Ausgabe: C=$\begin{pmatrix} c_{11} & \dots & c_{1n} \\ \vdots & & \vdots \\ c_{n1} & \dots & c_{nn} \end{pmatrix}$ mit $c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$, $1 \leq i,j \leq n$

Auftrag 5.1: Wie viele Multiplikationen und Additionen braucht es für die Berechnung von C? Wie viele Operationen sind es für allgemeines n, wie viele für n=2?

Auftrag 5.2: Wir betrachten für den Fall n=2 den alternativen Strassen-Algorithmus:

$A \cdot B = C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$ mit

$$c_{11} = m_1 + m_2 - m_4 + m_6$$
$$c_{12} = m_4 + m_5$$
$$c_{21} = m_6 + m_7$$
$$c_{22} = m_2 - m_3 + m_5 - m_7$$
$$m_1 = (a_{12} - a_{22}) \cdot (b_{21} + b_{22})$$
$$m_2 = (a_{11} + a_{22}) \cdot (b_{11} + b_{22})$$
$$m_3 = (a_{11} - a_{21}) \cdot (b_{11} + b_{12})$$
$$m_4 = (a_{11} + a_{12}) \cdot$$
$$m_5 = a_{11} \cdot (b_{12} - b_{22})$$
$$m_6 = a_{22} \cdot (b_{21} - b_{11})$$
$$m_7 = (a_{21} + a_{22}) \cdot b_{11}$$

Für einen Computer sind Multiplikationen viel aufwändiger als Additionen, so dass die benötigten Additionen vernachlässigt werden können: Warum ist der Strassen-Algorithmus effizienter als der naive Algorithmus? Wie viele Multiplikationen und Additionen braucht der Strassen Algorithmus für n=2?

Auftrag 5.3: Verifiziere, dass $c_{21} = m_6 + m_7$, dass heisst, das der Strassen-Algorithmus für c_{21} tatsächlich das gleiche berechnet wie der naive Algorithmus. Dies gilt auch für c_{11}, c_{12}, c_{22}.

Auftrag 5.4: Nun wollen wir den Strassen-Algorithmus für zwei $(n \times n)$-Matrizen betrachten, wobei n=2^k und $k \in N$:

M=$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$, N=$\begin{pmatrix} E & F \\ G & H \end{pmatrix}$ und

$$S_1 = (B - D)(G - H)$$
$$S_2 = (A + D)(E + H)$$
$$S_3 = (A - C)(E + F$$
$$S_4 = (A + B)H$$
$$S_5 = A(F - \mathrm{H})$$
$$S_6 = D(G - \mathrm{E})$$
$$S_7 = (C + \mathrm{D})\mathrm{E}$$

Wir berechnen: $M \cdot N = \begin{pmatrix} S_1 + S_2 - S_4 + S_6 & S_4 - S_5 \\ S_6 + S_7 & S_2 - S_3 + S_5 - S_7 \end{pmatrix}$

Verifiziere ähnlich wie in c), dass $S_6 + S_7 = CE + DG$.

(Wiederum berechnet der Strassen-Algorithmus dasselbe wie der naive Algorithmus.)

Auftrag 5.5: Sei T(n) die Anzahl der Operationen, welche der Strassen-Algorithmus für die Multiplikation zweier $(n \times n)$-Matrizen benötigt. Begründe warum die Rekurrenzgleichung $T(n) = 7 \cdot T\left(\frac{n}{2}\right) + \frac{9}{2}n^2$ gilt.

Auftrag 5.6: Bestimme eine explizite Form für T(n).

Auftrag 5.7: Die Anzahl Operationen vom naiven Algorithmus waren in O(n^3), die Anzahl Operationen vom Strassen-Algorithmus sind in O(n^x). Wie gross ist x?

[Bemerkung für Regula und Juraj: Ich verwende für Problem 5 die Methode „Divide and Conquer“, ich sehe nicht, wie eine einfache Induktion und eine Rekurrenzgleichung der Form T(n) = T(n-1) + … möglich ist. Über Hinweise dazu würde ich mich sehr freuen.]