Entwurf und Analyse   
von Algorithmen mit Hilfe der Induktion

Benedict Grupp & Kosta Kyriases

# Hinweise für die Lehrperson

Die vorliegende Ausarbeitung ist für die Einführung der Induktion als Methode zur Entwicklung und zur Effizienzanalyse von Algorithmen in einer Informatikklasse angelegt. Die Lernenden sollen anhand dieser Ausarbeitung und der Aufgabenstellungen in 3 Lektionen das Prinzip der Induktion sowie deren Nutzen zum Erstellen von effizienten Algorithmen eigenständig erarbeiten.

Folgende Vorannahmen und strukturelle Voraussetzungen unterliegen diesen Ausführungen:

**Sozialform:** Die Bearbeitung der Aufgaben sollte in einer Partnerarbeit oder als Gruppenarbeit erfolgen. Für die Erarbeitung des letzten Teils, in welchem man die Effizienz der Berechnung von Polynomen betrachtet, bietet es sich an, ein Gruppenpuzzle mit 3 Expertengruppen zu bilden, in welchen jeweils einer der drei Berechnungswege von jeweils einer Gruppe untersucht wird. Im Anschluss können alle Lernenden gemeinsam in der Stammgruppe im Kapitel *Untersuchung der Laufzeit bei der Berechnung von Polynomen* die Algorithmen auf ihre Effizienz vergleichen.

**Unterrichtsform:** Die vorliegende Ausarbeitung eignet sich als Vorlage für eine Selbst-erarbeitung durch die Lernenden. Dabei können die Lernenden in einem Stationenlernen durch die Inhalte geleitet werden, indem die einzelnen Kapitel (die Stationen) jeweils Teilaspekte fokussieren. Die Lehrperson kann die vorliegenden Materialien aber auch in Form eines mehrteiligen Online-Fragebogens mit hinterlegten Lösungen an die Lernenden abgeben. Hierbei sollten die Lernenden die Lösungen erst nach der Abgabe der eigenen Lösungen erhalten. Dennoch sollten sie nach jeder Teilaufgabe ein unmittelbares Feedback in Form einer (kommentierten) Lösung erhalten, um mögliche vorliegende Fehler bzw. Fehlkonzepte zu korrigieren. So wird ein erfolgreiches und vertieftes Verständnis durch Reflexion und Anwendung ermöglicht.

**Zeitvorgabe:** Die Lehrperson sollte drei Lektionen für die Erarbeitung einplanen. Am Ende des Kapitels *Untersuchung der Laufzeit bei der Berechnung von Polynomen* gibt es eine Zusatzaufgabe, welche den Lernenden zur Verfügung steht.

**Vorkenntnisse:**

Die Lernenden sollten über folgende mathematische Vorkenntnisse verfügen:

* Umgang mit Geraden in einem Koordinatensystem. Der Funktionsbegriff wird nicht benötigt.
* Umgang mit Polynomen.
* Termumformungen, insbesondere Potenzgesetze und Distributivgesetz.
* Umgang mit arithmetischen Reihen.

Aus der Informatik sollte bereits bekannt sein:

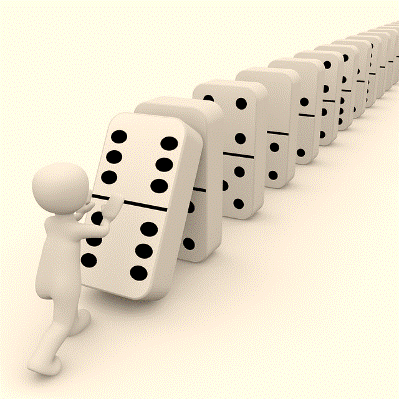
* Formalisierung von Algorithmen.

# Motivation

Um einen Algorithmus für ein bestimmtes Problem zu entwickeln, bedient man sich gerne bekannter Entwurfsmuster. Eines dieser Entwurfsmuster ist die Induktion.

Bei der **Induktion** nimmt man an, eine Lösung des betrachteten Problems für Eingaben der Grösse n-1 zu kennen und überlegt sich, wie man unter deren Verwendung eine Lösung für Eingaben der Grösse n bestimmen kann.   
Zusammen mit der Kenntnis der Lösung für Eingaben der Grösse n = 1 kann man eine Formel finden, welche für alle Eingaben der Grösse n eine Lösung liefert.

*Am* ***Dominoprinzip*** *kann man sich die Vorgehensweise gut veranschaulichen:*

*Stellt man die Steine so auf, dass gilt:*

1. *der erste Stein fällt beim Anstossen um,*
2. *wenn ein beliebiger Stein umfällt, dann stösst er den nachfolgenden Stein so an, dass dieser auch umfällt,*

*dann folgt daraus:  
Der erste Stein wird umgestossen, und nacheinander fallen alle Dominosteine um, da für jeden umgestossenen Stein der Nachfolger auch umfallen muss.*

* Wie man sich der Induktion bedienen kann, um einen Algorithmus zu entwickeln, werden wir uns im ersten Kapitel anschauen.
* Im zweiten Kapitel werden wir sehen, dass uns die Induktion zudem eine Methode zur Überprüfung der Effizienz eines Algorithmus liefert.

# Entwurf von Algorithmen mit Induktion

Betrachten wir folgende Fragestellung:

Zeichnet man eine beliebige Anzahl Geraden in eine zweidimensionale Ebene – wie viele Schnittpunkte haben die Geraden dann maximal?

Im Allgemeinen wird es schwerfallen, direkt eine Antwort für eine beliebige Anzahl Geraden zu finden.

Stattdessen beginnen wir mit Aussagen über eine **konkrete** Anzahl Geraden, beginnend mit dem einfachsten Fall - einer Geraden:

|  |  |  |
| --- | --- | --- |
| **Eine** Gerade hat **keine** Schnittpunkte: |  | **Zwei** Geraden haben **einen** Schnittpunkt: |
|  |  |  |
| Nimmt man eine **dritte** Gerade hinzu, kann diese höchstens **zwei** Schnittpunkte ergänzen: je einen mit den beiden vorhandenen Geraden: |  | Nimmt man eine **vierte** Gerade hinzu, kann diese höchstens **drei** Schnittpunkte ergänzen: je einen mit den drei vorhandenen Geraden: |
|  |  |  |

Man erkennt folgende Gesetzmässigkeit:

|  |
| --- |
| Die maximale Anzahl Schnittpunkte, welche in jedem Schritt hinzukommen, entspricht gerade der Anzahl Schnittpunkte der bereits vorhandenen Geraden. |

Sei S(n) die Anzahl Schnittpunkte bei n Geraden.

S(1) = 0

S(2) = S(1) + 1

Die maximale Anzahl Schnittpunkte bei n Geraden ist gleich der Anzahl Schnittpunkte bei n – 1 Geraden plus n – 1:

S(3) = S(2) + 2

S(4) = S(3) + 3

…

S(n) = S(n-1) + n-1

Eine induktive Formel zur Berechnung der Anzahl Schnittpunkte sieht also wie folgt aus:   
S(n) = S(n-1) + n-1 mit S(1) = 0

Diese Formel hat jedoch den Nachteil, dass man die Anzahl Schnittpunkte für n – 1 Geraden kennen muss, um die Anzahl Schnittpunkte für n Geraden zu berechnen:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Anzahl Geraden: **n** | 1 | 2 | 3 | 4 | 5 | … | n |
| Anzahl (maximal) hinzukommender Schnittpunkte | 0 | 1 | 2 | 3 | 4 | … | n-1 |
| Gesamtzahl (maximaler) Schnittpunkte bei n Geraden: **S(n)** | 0 | 1 | 3 | 6 | 10 | … | ? |

Um eine «bessere» Formel zu finden, welche die Anzahl Schnittpunkte für n Geraden direkt bestimmen kann, schauen wir uns wieder ein konkretes Beispiel an:

Will man die Anzahl Schnittpunkte für n = 6 Geraden berechnen, werden folgende Schritte durchgeführt:

S(6) = S(5) + 5

S(6) = S(4) + 4 + 5

S(6) = S(3) + 3 + 4 + 5

S(6) = S(2) + 2 + 3 + 4 + 5

S(6) = S(1) + 1 + 2 + 3 + 4 + 5

S(6) = 0 + 1 + 2 + 3 + 4 + 5

Wieder kann man eine Gesetzmässigkeit erkennen:

Die maximale Anzahl Schnittpunkte bei n Geraden ergibt sich aus der Summe der ganzen Zahlen von 0 bis n – 1.

Die zugehörige Formel sieht wie folgt aus:

S(n) = 0 + 1 + 2 + 3 + 4 + … + (n – 3) + (n – 2) + (n – 1)

Leider ist diese Formel nicht besonders effizient. Wir können zwar die Anzahl maximaler Schnittpunkte direkt bestimmen, jedoch erhöht sich der Rechenaufwand mit der Anzahl Geraden. Um auch dieses Problem zu lösen, bedienen wir uns der Erkenntnis eines wohlbekannten Mathegenies: Carl Friedrich Gauss.

Ein Bild, das Person, alt enthält.

Automatisch generierte BeschreibungDie Legende besagt, dass der Lehrer von Gauss (um die Schüler vielleicht längere Zeit ruhig zu beschäftigen) die Kinder aufforderte, die Zahlen von 1 bis 100 zu addieren. Doch seine Ruhe währte nicht lange, denn nach wenigen Augenblicken zeigte ihm der kleine Carl Friedrich das richtige Ergebnis. Wie gelang ihm dies so schnell?

Ganz einfach, indem er die Arbeit doppelt ausführte, aber auf eine sehr geschickte Art:

Carl Friedrich Gauss (1777 – 1855)

Schreibt man die Summanden in eine Zeile und nochmals in umgekehrter Reihenfolge eine Zeile darunter, sieht man, dass die Summe der Spalten jeweils n - 1 beträgt:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | … | n - 3 | n - 2 | n-1 |
| n - 1 | n - 2 | n - 3 | … | 2 | 1 | 0 |
| **n - 1** | **n - 1** | **n - 1** |  | **n - 1** | **n - 1** | **n - 1** |

n Spalten

Die Summe jeder Spalte ist n - 1. Da es n Spalten sind, ist die Summe der Zahlen beider Zeilen gleich *)*. Um die Summe der Zahlen einer Zeile zu ermitteln, wird das Ergebnis halbiert, und es ergibt sich die Formel:

Gauss zu Ehren wird die Formel auch «kleiner Gauss» genannt.

Diese Formel ist besonders effizient, weil der Rechenaufwand unabhängig von der Anzahl der Geraden immer gleich ist.

## Auftrag

In ein Rechteck werden Dreiecke wie folgt eingezeichnet:

|  |  |
| --- | --- |
| Im **ersten Schritt** wird ein Punkt in das Rechteck gezeichnet und mit den Ecken des Rechtecks verbunden. Es entstehen  4 Dreiecke. |  |
| In den **folgenden Schritten** wird jeweils ein weiterer Punkt innerhalb\* eines beliebigen Dreiecks gezeichnet und mit den Ecken dieses Dreiecks verbunden. So entstehen 3 weitere Dreiecke. |  |
| (\* Der Punkt darf nicht auf die Seite oder die Ecken des Dreiecks gelegt werden.) |  |

|  |  |
| --- | --- |
| Schritt 3: | Schritt 4: |
|  |  |

1. Zeichnen Sie die Dreiecke für Schritt 3 und Schritt 4 in die Bilder rechts ein (die Punkte sind bereits gesetzt).
2. Geben Sie an, wie viele Dreiecke in jedem Schritt vorhanden sind.
3. Wie viele Dreiecke wären mit 5 bzw. 6 Punkten vorhanden?
4. Finden Sie eine Formel, um zu berechnen, wie viele Dreiecke in einem Quadrat mit beliebig vielen n Punkten vorhanden sind.
5. Gegeben ist ein Rechteck, in das man 800 Dreiecke mit der obigen Methode einzeichnen will. Dabei muss man Punkte setzen bzw. Linien zwischen Punkten ziehen. Jeder dieser beiden Schritte bedarf eines Aufwands, der mit 1 angegeben wird. Welcher Aufwand entsteht, um die 800 Dreiecke in das Rechteck zu zeichnen?
6. Finden Sie eine Formel für den Aufwand für das Zeichnen von n Dreiecken, wobei n eine gerade Zahl sein muss.

## Weiterführende Aufgabe

Wie in der vorherigen Aufgabe werden immer weiter Punkte in ein Rechteck eingezeichnet, um nach dem gleichen Prinzip Dreiecke einzuzeichnen. Nun wird in jedem Schritt aber nicht nur ein weiterer Punkt hinzugefügt, sondern die Punkte werden mit jedem Schritt verdoppelt:

|  |  |
| --- | --- |
| Schritt 3: | Schritt 4 |
|  |  |

1. Zeichnen Sie die Dreiecke für Schritt 3 und Schritt 4 in die Bilder rechts ein (die Punkte sind bereits gesetzt).
2. Geben Sie an, wie viele Dreiecke in jedem Schritt vorhanden sind.
3. Wie viele Dreiecke wären mit 16 bzw. 32 Punkten vorhanden?
4. Finden Sie eine Formel, um zu berechnen, wie viele Dreiecke in einem Quadrat nach n Iterationen vorhanden sind.

# Effizienz von Algorithmen – Die Laufzeitanalyse

Hat man einen **Algorithmus** gefunden, stellt sich die Frage, ob dieser **effizient** ist, also ob er z.B. in einer akzeptablen Zeit ein Ergebnis liefert. Über die Effizienzprüfung wird also die «Laufzeit» geprüft. Die Bestimmung der Laufzeit eignet sich zudem für den Vergleich von Algorithmen, welche dasselbe Problem lösen. Welcher Algorithmus erledigt die Aufgabe am schnellsten und kostengünstigsten? Mit diesen Fragen beschäftigt sich die **Laufzeitanalyse**.

## Wie bestimmt man die Laufzeit?

Eine naheliegende Möglichkeit wäre, die Laufzeit als **Dauer der Durchführung** zu definieren. Man könnte also die Zeitspanne messen, in welcher der Computer den Algorithmus ausführt. Dies erweist sich jedoch als problematisch, da die Zeitspanne auch von Faktoren wie z.B. der Taktfrequenz der CPU, der Speichergrösse des Rechners oder der verwendeten Programmiersprache beeinflusst wird und die das Ergebnis verfälschen können. Aus diesem Grund gibt man in der Informatik Laufzeiten von Algorithmen nicht in Zeiteinheiten an.

Stattdessen zählt man die Anzahl Operationen (Addition, Multiplikation …), welche vom Algorithmus benötigt werden, und definiert die Laufzeit als die Anzahl benötigter Elementaroperationen bei einer bestimmten Eingabelänge n:

Laufzeit = benötigte Elementaroperationen bei einer  
bestimmten Eingabelänge n

Wir werden sehen, dass uns auch hier der induktive Ansatz zum Ziel führt.

In der folgenden Aufgabe werden wir uns ein Problem anschauen, welches mit drei verschiedenen Algorithmen gelöst werden kann:

Gegeben ist ein Polynom ().   
Berechnen Sie den Wert des Polynoms für gegebene und .

**Beispiel:**   
Es soll der Wert des Polynoms für berechnet werden. Setzt man in das Polynom ein, erhält man .

**Aufgabe:**

1. Wie viele Multiplikationen braucht man mindestens, um die Potenzen im Polynom zu berechnen? Füllen Sie die Tabelle aus (die erste Zeile ist bereits als Orientierung gegeben):

|  |  |  |
| --- | --- | --- |
| **Potenzen** | **Berechnung** | **Anzahl Multiplikationen** |
|  |  | Eine Multiplikation |
|  |  |  |
|  |  |  |

Wie viele Multiplikationen haben Sie insgesamt benötigt, um die Tabelle zu füllen?

Notieren Sie, wie viele Additionen und Multiplikationen Sie benötigen, um für   
 zu berechnen.

1. Wie könnte man die Potenzen per Induktion berechnen, indem man immer auf die nächstkleinere Potenz zurückgreift:

Man könnte also die n-te Potenz induktiv mit Hilfe der Potenz berechnen, indem man die folgende Gesetzmässigkeit nutzt: . Hier

Füllen Sie nun mit diesem Vorgehen erneut die Tabelle und geben Sie an, wie viele Multiplikationen Sie benötig haben:

|  |  |  |
| --- | --- | --- |
| **Potenzen** | **Berechnung** | **Anzahl Multiplikationen** |
|  |  | Eine Multiplikation |
|  |  |  |
|  |  |  |

Wie viele Multiplikationen haben Sie insgesamt benötigt, um die Tabelle zu füllen?

Wie viele Multiplikationen würden Sie insgesamt benötigen, wenn die Tabelle bis zur Potenz gehen würde?

Notieren Sie, wie viele Additionen und Multiplikationen Sie benötigen, um für   
 zu berechnen.

## Das Horner-Schema als Algorithmus zum Berechnen von Polynomen

Man kann Polynome auch mit Hilfe des Horner-Schemas notieren. Hierbei wird ein Polynom () in folgenden Term umgeformt:

()

Um die Klammerung übersichtlicher zu machen, wurde sie farblich unterschieden.

**Bsp.:** wird umgeformt zu durch folgende Schritte:

|  |  |
| --- | --- |
| Umformungsschritt |  |
|  |  |
| faktorisieren |  |
|  |  |
| faktorisieren |  |
|  |  |

1. Formen Sie das Polynom aus der Einstiegsaufgabe mit Hilfe des Horner-Schemas um.
2. Berechnen Sie den Wert des Polynoms mit Hilfe des Horner-Schemas für . Notieren Sie hierbei, wie viele Additionen/Subtraktionen und wie viele Multiplikationen Sie benötigt haben.
3. Beim Berechnen des Polynoms mit Hilfe des Horner-Schemas kann man wie folgt vorgehen:

Beschreiben Sie in eigenen Worten, wie hier das Prinzip der Induktion angewendet wird.

## Untersuchung der Laufzeit bei der Berechnung von Polynomen

Abschliessend wollen wir die Laufzeit, d.h. die Anzahl der Berechnungsschritte beim Berechnen von Polynomen für ein bestimmtes betrachten. Hierfür haben wir nun drei Wege zur Auswahl:

1. Das Berechnen von Polynomen, indem man die Potenzen nicht effizient berechnet, d.h., wird durch berechnet.
2. Das Berechnen von Polynomen, indem man die Potenzen effizient berechnet, d.h., wird durch berechnet.
3. Das Berechnen mit Hilfe des *Horner-Schemas*.
4. In den vorhergehenden Aufgaben haben Sie die Anzahl Additionen und Multiplikationen notiert, die Sie beim Berechnen für für verschiedene benötigt haben. Notieren Sie in der Tabelle den Aufwand:

|  |  |  |  |
| --- | --- | --- | --- |
| Anzahl Operationen | 1. Weg | 1. Weg | Horner-Schema |
| Addition |  |  |  |
| Multiplikation |  |  |  |

1. In der folgenden Tabelle untersuchen wir nun die Anzahl Berechnungsschritte für Polynome verschiedenen Grades, beginnend mit Grad 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Grad | Polynom | 1. Weg | 1. Weg | Horner-Schema |
| n = 1 |  | 1 Multiplikation,  1 Addition,  **Gesamt:  2 Operationen** | 1 Multiplikation,  1 Addition,  **Gesamt: 2 Operationen** | 1 Multiplikation,  1 Addition,  **Gesamt: 2 Operationen** |
| n = 2 |  |  |  |  |
| n = 3 |  |  |  |  |
| n = 4 |  |  |  |  |

1. Abschliessend wollen wir die Laufzeit der drei betrachteten Wege für Polynome von Grad n bestimmen. Welcher der drei Algorithmen benötigt am wenigsten Rechenoperationen?

**Hinweis:** Die Anzahl der Multiplikationen für den 1. Weg ist bereits eingetragen. Diese kann mit Hilfe des «kleinen Gauss» berechnet werden, denn mit jeder Erhöhung des Grades des Polynoms um den Wert 1 nimmt die Anzahl der Multiplikationen um den Wert des höchsten Grades zu, was der Gesetzmässigkeit zur Berechnung der Schnittpunkte von Geraden aus Kapitel 1 entspricht.

|  |  |  |  |
| --- | --- | --- | --- |
| Polynom | 1. Weg | 2. Weg | Horner-Schema |
|  | Multiplikation,  n Additionen,  **Gesamt:**  **Operationen** |  |  |

**Zusatzaufgabe:**

Untersuchen Sie die Effizienz der drei Algorithmen für die Berechnung des Polynoms   
 für ein . Welche Aussage über die Effizienz der Algorithmen kann man treffen?

|  |  |  |  |
| --- | --- | --- | --- |
| Polynom | 1. Weg | 2. Weg | Horner-Schema |
|  | Multiplikation,  n Additionen,  **Gesamt:**  **Operationen** |  |  |

# Lösungen

Kapitel 1

Auftrag:

1. und 2.

|  |  |  |  |
| --- | --- | --- | --- |
| Schritt 1: 1 Punkt, 4 Dreiecke | Schritt 2: 2 Punkte, 6 Dreiecke | Schritt 3: 3 Punkte, 8 Dreiecke | Schritt 4: 4 Punkte, 10 Dreiecke |
|  |  | Ein Bild, das Text, Zubehör enthält.  Automatisch generierte Beschreibung |  |

1. Mit 5 Punkten wären 12 und mit 6 Punkten wären 14 Dreiecke vorhanden.
2. A(n)=2n+2
3. und 6. Für 800 Dreiecke benötigt man n=399 Iterationen. Pro gesetzten Punkt werden 3 neue Linien gezogen, nur beim ersten Punkt werden 4 Linien gezogen. Damit ist in jedem Iterationsschritt der Aufwand 4, nur beim ersten ist der Aufwand um eins erhöht, also 5. Somit ist der Aufwand B(399)=4\*399+1=1597.

Weiterführende Aufgabe:

1. und 2.

|  |  |  |  |
| --- | --- | --- | --- |
| Schritt 1: 1 Punkt, 4 Dreiecke | Schritt 2: 2 Punkte, 6 Dreiecke | Schritt 3: 4 Punkte, 10 Dreiecke | Schritt 4: 8 Punkte, 18 Dreiecke |
|  |  |  |  |

1. Mit 16 Punkten wären 34 und mit 32 Punkten wären 66 Dreiecke vorhanden.
2. A(n)=2^n+2

Kapitel 2

Aufgabe

a)

|  |  |  |
| --- | --- | --- |
| **Potenzen** | **Berechnung** | **Anzahl Multiplikationen** |
|  |  | Eine Multiplikation |
|  |  | Zwei Multiplikationen |
|  |  | Drei Multiplikationen |
| Insgesamt sechs Multiplikationen | | |

Man benötigt 10 Multiplikationen und 4 Additionen.

b)

|  |  |  |
| --- | --- | --- |
| **Potenzen** | **Berechnung** | **Anzahl Multiplikationen** |
|  |  | Eine Multiplikation |
|  |  | Eine Multiplikation |
|  |  | Eine Multiplikation |
| Insgesamt drei Multiplikationen | | |

Würde die Tabelle bis zur Potenz gehen, würde man neun Multiplikationen benötigen.

Man benötigt 7 Multiplikationen und 4 Additionen, um zu berechnen.

Das Horner-Schema

a)

Lösung:

b)

Lösung:   
Man benötigt 4 Multiplikationen und 4 Additionen.

c)

Man berechnet mit , wobei der Koeffizient der nächstkleineren Potenz ist und das Argument, für welches man den Wert des Polynoms berechnet. Somit greift man für das Berechnen immer auf ein Teilergebnis zurück.

Untersuchung der Laufzeit

a)

|  |  |  |  |
| --- | --- | --- | --- |
| Anzahl Operationen | 1. Weg | 1. Weg | Horner-Schema |
| Addition | 4 | 4 | 4 |
| Multiplikation | 10 | 7 | 4 |

b)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Grad | Polynom | 1. Weg | 1. Weg | Horner-Schema |
| n = 1 |  | 1 Multiplikation,  1 Addition,  **Gesamt:  2 Operationen** | 1 Multiplikation,  1 Addition,  **Gesamt:  2 Operationen** | 1 Multiplikation,  1 Addition,  **Gesamt:  2 Operationen** |
| n = 2 |  | 3 Multiplikationen,  2 Additionen,  **Gesamt:  5 Operationen** | 3 Multiplikationen,  2 Additionen,  **Gesamt:  5 Operationen** | 2 Multiplikationen,  2 Additionen,  **Gesamt:  4 Operationen** |
| n = 3 |  | 6 Multiplikationen, 3 Additionen,  **Gesamt:  9 Operationen** | 5 Multiplikationen,  3 Additionen,  **Gesamt:  8 Operationen** | 3 Multiplikationen,  3 Additionen,  **Gesamt:  6 Operationen** |
| n = 4 |  | 10 Multiplikationen, 4 Additionen,  **Gesamt:  14 Operationen** | 7 Multiplikationen,  4 Additionen,  **Gesamt:  11 Operationen** | 4 Multiplikationen,  4 Additionen,  **Gesamt:  8 Operationen** |

c)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Grad | Polynom | 1. Weg | 1. Weg | Horner-Schema |
| n = 1 |  | Multiplikation,  n Additionen,  **Gesamt:**  **Operationen** | Multiplikation,  n Addition,  **Gesamt:  3n-1 Operationen** | n Multiplikation,  n Addition,  **Gesamt:  2n Operationen** |

Zusatzaufgabe:

|  |  |  |
| --- | --- | --- |
| 1. Weg | 1. Weg | Horner-Schema |
| Multiplikation,  n Additionen,  **Gesamt:**  **Operationen** | n-1 Multiplikationen, n Additionen,  **Gesamt: 2n-1 Operationen** | n-1 Multiplikationen, n Additionen,  **Gesamt: 2n-1 Operationen** |

Im Fall, dass die Koeffizienten eines Polynoms alle 1 sind, bringt das Horner-Schema keine kürzere Laufzeit, da ebenso viele Operationen anfallen wie bei der effizienten Berechnung von Potenzen.