

# INFORMATIK

Klasse: 3.Klasse

Erstellt von: GRUBER ELIAS

Erstellt am: 21. August 2018

## KURZFASSUNG

### **Einleitung**

Hier werden die wesentlichen Punkte des Informatik-Unterrichts der 3.Klasse erwähnt. Das Skript sollte als Lernunterlage für die Lehrsabschlussprüfung dienen und das Erlernen der einzelnen Themen erleichtern.

Weiteres sind alle Angaben ohne Gewähr und die einzelnen Themen sind auf das Wesentliche heruntergebrochen. Die Weitergabe an Personen, die nicht von GRUBER ELIAS genehmigt wurde, ist nicht gestattet.

## INHALTSVERZEICHNIS

1. ALGORITHMEN GRUNDLAGEN/BEGRIFFE	4
2. ALGORITHMUS DARSTELLUNGSMÖGLICHKEITEN	5
3. DIVERSE ALGORITHMEN	6
3.1 EUKLIDISCHER ALGORITHMUS	6
4. DATENSTRUKTUREN	7
4.1 ARTEN VON DATENSTRUKTUREN	7
5. LINEARE LISTEN	8
6. SORTIERVERFAHREN	9
6.1 BUBBLE SORT	9
6.2 MERGE SORT	10
6.3 QUICK SORT	10
6.4 SELECTION SORT	11
6.5 INSERTION SORT	11
7. DATENSTRUKTUR BAUM	12
7.1 BINÄRBÄUME	12
7.2 TRAVERSIERUNG	13
7.3 AVL	14
7.4 B-Baum/2-3-4 Baum/ROT-SCHWARZ Baum	14
8. SUCHVERFAHREN	15
8.1 SEQUENTIELLE SUCHE	15
8.2 BINÄRE SUCHE	15
8.2 BINÄRER SUCHBAUM	15
9. GRAPH	16
9.1 DIJKSTRA	16
9.2 WARSHALL	16
9.3 GRAPHDATENBANK	16
9.4 BACKTRACKING	17
9.5 ADJAZENZMATRIX	17

---

## 1. Algorithmen Grundlagen/Begriffe

Das Wort Algorithmus kommt aus dem arabischen Raum und man bezeichnet es als eine systematische, logische Regel oder Vorgehensweise, die zur Lösung eines vorliegenden Problems führt.

### **Was ist ein Algorithmus?**

Ist eine Beschreibung, wie eine Bestimmte Aufgabe gelöst wird. Ein Algorithmus enthält eine Folge von elementaren Schritten die in der angegebenen Reihenfolge ausgeführt werden müssen.

### **Was ist ein Programm?**

Ist ein Algorithmus in einer dem Computer verständlichen Sprache (Programmiersprache).

### **Was ist ein Prozess?**

Ist ein Vorgang der Abarbeitung eines Algorithmus

### **Was ist ein Struktogramm/Flussdiagramm?**

Ist eine grafische Darstellung eines Algorithmus.

### **Was ist ein rekursiver Algorithmus?**

Ist ein Algorithmus der sich selbst als Einzelschritt enthält. (Bsp. Fakultät -  $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$ )

### **Welche Anforderungen gibt es an einen Algorithmus?**

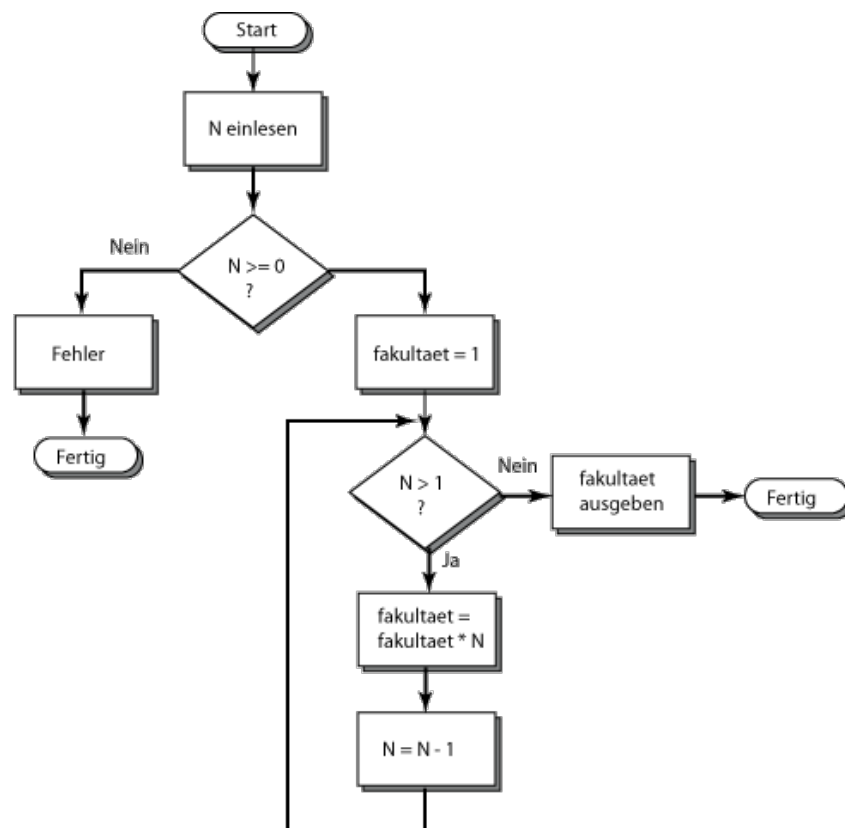
- ◆ Eindeutigkeit
- ◆ Endlichkeit / Zielgerichtetheit
- ◆ Universalität
- ◆ Korrektheit
- ◆ Determinismus

## 2. Algorithmus Darstellungsmöglichkeiten

Ein Algorithmus kann auf verschiedenste Weise dargestellt werden. Die optische Darstellung erfolgt mittels Struktogramm, Flußdiagramm.

### Copyshop

Zählerstand alt eingeben			
Zählerstand neu eingeben			
Verbrauch := Zählerstand neu - Zählerstand alt			
Verbrauch =			
1 .. 10	11 .. 50	51 .. 100	sonst
Preis := 0.12	Preis := 0.07	Preis := 0.06	Preis := 0.05
NVP := Verbrauch * Preis			
MwST := NVP * 0.16			
BVP := NVP + MwSt			
Alle Werte ausgeben			



---

### 3. Diverse Algorithmen

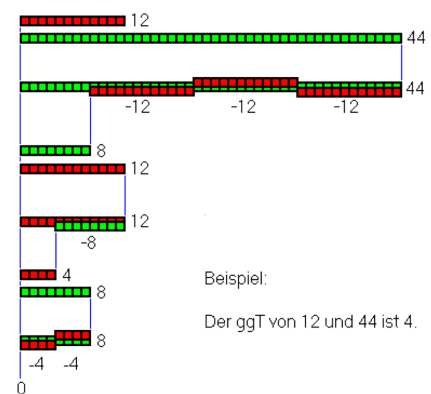
#### 3.1 Euklidischer Algorithmus

Durch den euklidischen Algorithmus kann der größte gemeinsame Teiler (ggT) von zwei Zahlen berechnet werden. Hierfür ist folgende Vorgehensweise einzuhalten:

Man teilt die größere Zahl durch die kleinere. Sollte die Division direkt aufgehen und kein Rest vorhanden sein, ist das Ergebnis schon der ggT. Ansonsten wird der Rest zum neuen Divisor und der alte Divisor zum Dividenten. Man setzt das so lange fort, bis man auf den letztmöglichen ggT kommt und kein Rest mehr enthalten ist.

#### Beispiele:

544	:	391 = 1;	Rest 153
391	:	153 = 2;	Rest 85
153	:	85 = 1;	Rest 68
85	:	68 = 1;	Rest 17
68	:	17 = 4;	Rest 0



---

## 4. Datenstrukturen

In der Informatik und Softwaretechnik ist eine Datenstruktur ein Objekt zur Speicherung und Organisation von Daten. Es handelt sich um eine Struktur, weil die Daten in einer bestimmten Art und Weise angeordnet und verknüpft werden, um den Zugriff auf sie und ihre Verwaltung effizient zu ermöglichen.

### **Allgemeine Eigenschaften von Daten:**

Es existieren Basisdaten wie Zeichen, Wahrheitswerte (true,false), ganze Zahlen oder Gleitpunktzahlen. Daten können Beziehungen untereinander haben, wie z.B. Listen, hierarchische Datenstrukturen wie Bäume.

### **Basis Datentypen:**

char:	Menge der Zeichen
int:	Menge der ganzen Zahlen, die im Rechner darstellbar sind
float:	Menge der darstellbaren Gleitkommazahlen mit einfacher Genauigkeit
double:	Menge der darstellbaren Gleitkommazahlen mit doppelter Genauigkeit
Array:	Zusammenfassung von zusammengehörigen Daten des gleichen Typs.

### 4.1 Arten von Datenstrukturen

#### **Datensatz**

- ◆ Wird auch Tupel oder Record genannt
- ◆ Gehören zu den einfachsten Datenstrukturen
- ◆ Sie verkörpern Werte, die andere Werte enthalten, in einer fest definierten Anzahl und Folge

#### **Array**

- ◆ Einfachste verwendete Datenstruktur
- ◆ Mehrere Variablen vom selben Basisdatentyp gespeichert

#### **Verkettete Liste**

- ◆ Zur dynamischen Speicherung von beliebig vielen Objekten
- ◆ Jedes Element beinhaltet einen Zeiger auf das nächste Element

#### **Stapelspeicher (Stack)**

- ◆ Funktioniert nach dem LIFO-Prinzip (Last-In-First-Out)
- ◆ Kann eine beliebige Anzahl von Objekten gespeichert werden
- ◆ Objekte können nur in umgekehrter Reihenfolge gelesen werden
  - push = Objekt ablegen
  - pop = letzte gespeicherte Objekt vom Stapel nehmen und entfernen
  - top = oberste Element des Stacks, ohne das es entfernt wird

---

### **Warteschlange (Queue)**

- ◆ Funktioniert nach dem FIFO-Prinzip (First-In-First-Out)
- ◆ Es kann eine beliebige Anzahl von Objekten gespeichert werden

### **Bäume**

- ◆ Hierarchische Struktur
- ◆ Die durch die Hierarchie vorgegebenen Objekte nennt man Knoten
- ◆ Verweise werden Kanten genannt

## *5. Lineare Listen*

### **Was ist eine lineare Liste?**

Bist auf das letzte Element hat jedes Element genau einen Nachfolger.

1. EL → 2. EL → 3.EL → 4.EL → letztes EL

Listen können in 3 verschiedenen Formen gespeichert werden.

- ◆ **sequentielle Speicherung**
  - ◆ gleiche Länge je Datenelement → direkter Zugriff auf Element (z.B. Array)
  - ◆ unterschiedliche Länge je Datenelement → ganze Liste muss von vorne durchsucht werden (z.B. Texte durchsuchen)
- ◆ **gekettete Speicherung**
  - ◆ einfach verkettete Liste
  - ◆ mehrfach verkettete Liste
  - ◆ doppelt verkettete Liste
  - ◆ kreisförmig verkettete Liste
- ◆ **indizierte Speicherung**
  - ◆ zusätzlich zu den Datenelementen werden Indextabellen bzw. Indexdateien gespeichert, welche die Suchbegriffe (Schlüssel,Key) und einen Zeiger auf das Datenelement enthalten.



---

## 6. Sortierv Verfahren

### **Was ist ein Sortierv Verfahren?**

Ein Sortierv Verfahren ist ein Algorithmus, der dazu dient, ein Tupel (i. A. ein Array) zu sortieren.

Zum Beispiel die lexikographische Ordnung von Zeichenketten oder die numerische Ordnung von Zahlen.

Es gibt verschiedene Sortierv Verfahren, die unterschiedlich effizient arbeiten bezüglich der Zeitkomplexität (Anzahl der nötigen Operationen) sowie der Platzkomplexität (zusätzlich zum Eingabe-Array benötigter weiterer Speicherplatz). Die Zeitkomplexität hängt bei einigen Sortierv Verfahren von der anfänglichen Anordnung der Werte im Array ab, man unterscheidet dann zwischen Best Case (bester Fall), Average Case (Durchschnittsverhalten) und Worst Case (schlechtester Fall ~ die Werte sind „maximal ungünstig vorgeordnet“).

### **Stabiles und instabiles Suchverfahren**

Ein stabiles Sortierv Verfahren ist ein Sortieralgorithmus, der die Reihenfolge der Datensätze, deren Sortierschlüssel gleich sind, bewahrt.

Wenn bspw. eine Liste alphabetisch sortierter Personendateien nach dem Geburtsdatum neu sortiert wird, dann bleiben unter einem stabilen Sortierv Verfahren alle Personen mit gleichem Geburtsdatum alphabetisch sortiert.

#### *6.1 Bubble Sort*

### **Bezeichnung/Definition: paarweises Austauschen**

Bubblesort (auch Sortieren durch Aufsteigen oder Austauschsortieren) ist ein Algorithmus, der vergleichsbasiert eine Liste von Elementen sortiert. Dieses Sortierv Verfahren arbeitet in-place, sortiert stabil und hat eine Laufzeit von im schlimmsten (Worst- Case) wie auch im durchschnittlichen Fall (Average-Case).

### **Prinzip:**

In der Bubble-Phase wird die Eingabe-Liste von links nach rechts durchlaufen. Dabei wird in jedem Schritt das aktuelle Element mit dem rechten Nachbarn verglichen. Falls die beiden Elemente das Sortierkriterium verletzen, werden sie getauscht. Am Ende der Phase steht bei auf- bzw. absteigender Sortierung das größte bzw. kleinste Element der Eingabe am Ende der Liste.

---

## 6.2 Merge Sort

### **Bezeichnung/Definition: sortieren durch Verschmelzen/Vertauschen**

Mergesort (von englisch merge ‚verschmelzen‘ und sort ‚sortieren‘) ist ein stabiler Sortieralgorithmus, der nach dem Prinzip teile und herrsche (divide and conquer) arbeitet.

#### **Prinzip:**

Mergesort betrachtet die zu sortierenden Daten als Liste und zerlegt sie in kleinere Listen, die jede für sich sortiert werden. Die sortierten kleinen Listen werden dann im Reißverschlussverfahren zu größeren Listen zusammengefügt, bis wieder eine sortierte Gesamtliste erreicht ist.

## 6.3 Quick Sort

### **Bezeichnung/Definition: sortieren durch Teilen**

Quicksort (englisch quick, deutsch ‚schnell‘ und to sort ‚sortieren‘) ist ein schneller, rekursiver, nicht-stabiler Sortieralgorithmus, der nach dem Prinzip Teile und herrsche (lateinisch Divide et impera!, englisch divide and conquer) arbeitet. Der Algorithmus hat den Vorteil, dass er über eine sehr kurze innere Schleife verfügt (was die Ausführungsgeschwindigkeit stark erhöht) und ohne zusätzlichen Speicherplatz auskommt.

#### **Prinzip:**

Zunächst wird die zu sortierende Liste in zwei Teillisten („linke“ und „rechte“ Teilliste) getrennt. Dazu wählt Quicksort ein sogenanntes Pivotelement aus der Liste aus. Alle Elemente, die kleiner als das Pivotelement sind, kommen in die linke Teilliste, und alle, die größer sind, in die rechte Teilliste. Die Elemente, die gleich dem Pivotelement sind, können sich beliebig auf die Teillisten verteilen. Nach der Aufteilung sind die Elemente der linken Liste kleiner oder gleich den Elementen der rechten Liste.

Anschließend muss man also nur noch jede Teilliste in sich sortieren, um die Sortierung zu vollenden. Dazu wird der Quicksort-Algorithmus jeweils auf der linken und auf der rechten Teilliste ausgeführt. Jede Teilliste wird dann wieder in zwei Teillisten aufgeteilt und auf diese jeweils wieder der Quicksort-Algorithmus angewandt, und so fort. Diese Selbstaufrufe werden als Rekursion bezeichnet. Wenn eine Teilliste der Länge eins oder null auftritt, so ist diese bereits sortiert und es erfolgt der Abbruch der Rekursion.

---

## 6.4 Selection Sort

### **Bezeichnung/Definition: passendes Element aussuchen**

Selectionsort ist ein einfacher („naiver“) Sortieralgorithmus, der in-place arbeitet und in seiner Grundform instabil ist, wobei er sich auch stabil implementieren lässt. Die Komplexität von Selectionsort ist (Landau-Notation). Alternative Bezeichnungen des Algorithmus sind MinSort (von Minimum) bzw. MaxSort (von Maximum), Selectsort oder ExchangeSort (AustauschSort).

### **Prinzip:**

Sei  $S$  der sortierte Teil des Arrays (vorne im Array) und  $U$  der unsortierte Teil (dahinter). Am Anfang ist  $S$  noch leer,  $U$  entspricht dem ganzen (restlichen) Array. Das Sortieren durch Auswählen läuft nun folgendermaßen ab:

Suche das kleinste Element in  $U$  und vertausche es mit dem ersten Element von  $U$  (= das erste Element nach  $S$ ).

## 6.5 Insertion Sort

### **Bezeichnung/Definition: passendes Element einfügen**

Insertionsort ist ein einfaches stabiles Sortierverfahren (d. h. die Reihenfolge von Elementen mit gleichem Schlüsselwert bleibt unverändert). Es ist leicht zu implementieren, effizient bei kleinen oder bereits teilweise sortierten Eingabemengen. Außerdem benötigt Insertionsort keinen zusätzlichen Speicherplatz, da der Algorithmus in-place arbeitet. Ein weiterer Vorteil besteht darin, dass Insertionsort als Online-Algorithmus eingesetzt werden kann.

### **Prinzip:**

Das Vorgehen ist mit der Sortierung eines Spielkartenblatts vergleichbar. Am Anfang liegen die Karten des Blatts verdeckt auf dem Tisch. Die Karten werden nacheinander aufgedeckt und an der korrekten Position in das Blatt, das in der Hand gehalten wird, eingefügt. Um die Einfügestelle für eine neue Karte zu finden, wird diese sukzessive (von links nach rechts) mit den bereits einsortierten Karten des Blattes verglichen. Zu jedem Zeitpunkt sind die Karten in der Hand sortiert und bestehen aus den zuerst vom Tisch entnommenen Karten.

---

## 7. Datenstruktur Baum

### **Definition:**

Unter Bäumen in der Informatik versteht man eine bestimmte Art dynamischer Datenstrukturen, die dazu verwendet werden Daten hierarchisch speichern zu können. Dynamisch deshalb weil die Anzahl der Elemente nicht von Vorhinein begrenzt ist, sondern jedesmal wenn ein neues Element gespeichert werden soll Speicherplatz dafür reserviert werden kann und deshalb die Anzahl der Elemente die gespeichert werden können nicht von der Datenstruktur sondern der Größe des zur Verfügung stehenden Speichers abhängig ist.

### **Definition erweitert:**

Zettel anfügen (Datenstruktur Baum Übersicht)

### **Eigenschaften:**

Ein Baum wird durch folgende Eigenschaften beschrieben:

- ◆ Prinzipiell besteht ein Baum aus sogenannten Knoten und Kanten .
- ◆ Die Daten werden in Knoten gespeichert, diese Knoten sind mittels Kanten verbunden.
- ◆ Es gibt einen Knoten, der keinen Vorgänger hat, dieser wird Wurzel genannt.  
Die Wurzel stellt sozusagen den Startpunkt im Baum dar.
- ◆ Alle anderen Knoten haben genau einen Vorgänger.
- ◆ Knoten können beliebig viele Nachfolger haben, Vorgänger und Nachfolger sind mittels Kanten mit dem Knoten verbunden.

### 7.1 Binärbäume

#### **Definition:**

Binärbäume sind Bäume mit den besonderen Eigenschaften, dass jeder Knoten maximal 2 Nachfolger hat (den linken und den rechten Teilbaum). Der ganze Baum ist so nach einem Schlüssel sortiert, dass alle Knoten die sich im linken Teilbaum befinden niedrigwertiger sind und alle Knoten die sich im rechten Teilbaum befinden höherwertiger sind als der betrachtete Knoten. Knoten die keinen Nachfolger haben werden Blattknoten genannt.

#### **Vorteile:**

- ◆ Schnelle Suche
- ◆ für rekursive Arbeitsweise geeignet (jeder Knoten kann so behandelt werden als wäre er die Wurzel eines ganzen Binärbaumes)

#### **Nachteile:**

- ◆ Wenn oft Datenelemente eingefügt und wieder gelöscht werden kann der Baum degenerieren.

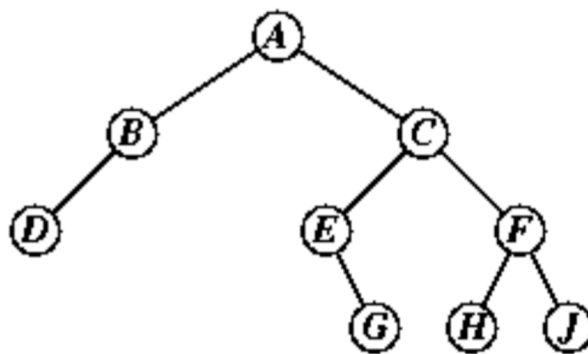
---

## 7.2 Traversierung

Ziel der Traversieren ist es, Bäume in einer Liste darstellen zu können. Es gibt drei prinzipielle Möglichkeiten, einen binären Baum zu traversieren:

- ♦ **preorder** (Besuche die Wurzel → Traversiere den linken Teilbaum → Traversiere den rechten Teilbaum)
- ♦ **inorder** (Traversiere den linken Teilbaum → Besuche die Wurzel → Traversiere den rechten Teilbaum)
- ♦ **postorder** (Traversiere den linken Teilbaum → Traversiere den rechten Teilbaum → Besuche die Wurzel)

**Beispiel:**



Folgende Reihenfolgen ergeben sich, wenn die Knoten entsprechend den vorgestellten Traversen besucht werden:

---

*preorder*

A B D C E G F H J

---

*inorder*

D B A E G C H F J

---

*postorder*

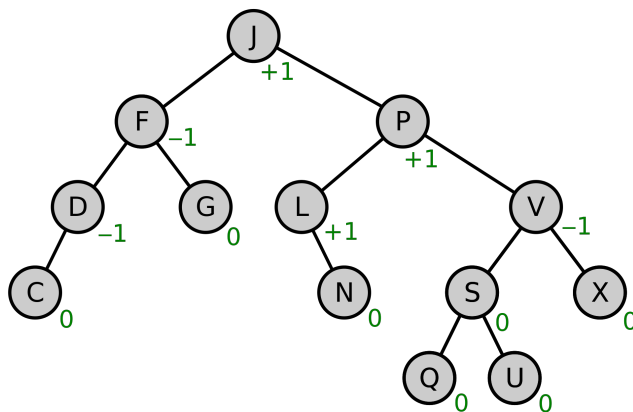
D B G E H J F C A

---

---

### 7.3 AVL

Der AVL-Baum ist eine Datenstruktur in der Informatik, und zwar ein binärer Suchbaum mit der zusätzlichen Eigenschaft, dass sich an jedem Knoten die Höhe der beiden Teilbäume um höchstens eins unterscheidet.



### 7.4 B-Baum/2-3-4 Baum/Rot-Schwarz Baum

#### **B-Baum:**

Anwendung finden B-Bäume unter anderem bei Datenbanksystemen, die mit sehr großen Datenmengen umgehen müssen, von denen nur ein Bruchteil gleichzeitig in den Hauptspeicher eines Rechners passt. In einem B-Baum kann ein Knoten - im Unterschied zu Binärbäumen - mehr als 2 Kind- Knoten haben.

#### **2-3-4-Baum:**

Ein 2-3-4-Baum ist in der Informatik eine Datenstruktur, genauer ein B- Baum des Verzweigungsgrades 2, das heißt, er ist ein Baum, in dem jeder Knoten zwei, drei oder maximal vier Kinder besitzt und entsprechend ein, zwei oder maximal drei Datenelemente speichert, die nach dem gewählten Ordnungskriterium aufsteigend sortiert sind. Er stellt damit zugleich einen speziellen balancierten Suchbaum dar.

Der 2-3-4-Baum wird häufig zur Speicherung großer Datenmengen verwendet.

---

### **Rot-Schwarz Baum:**

Ein Rot-Schwarz-Baum ist eine Datenstruktur vom Typ Binärer Suchbaum, die „sehr schnellen“ Zugriff auf die in ihr gespeicherten Werte garantiert. Die schnellen Zugriffszeiten auf die einzelnen im Rot-Schwarz-Baum gespeicherten Elemente werden durch drei Forderungen erreicht, die zusammen garantieren, dass ein Rot-Schwarz-Baum immer balanciert ist.

Zusätzlich zu den Eigenschaften des binären Suchbaums hat jeder Knoten des Rot-Schwarz- Baums ein weiteres Attribut, genannt Farbe, mit zwei Werten, genannt rot und schwarz. Diese Einfärbung hat die drei folgenden Forderungen zu erfüllen:

1. Alle externen Blatt-Knoten sind schwarz.
2. Ist ein Knoten rot, so sind beide Kinder schwarz.
3. Jeder Pfad von einem gegebenen Knoten zu seinen Blatt knoten enthält die gleiche Anzahl schwarzer Knoten.

## 8. Suchverfahren

### 8.1 Sequentielle Suche

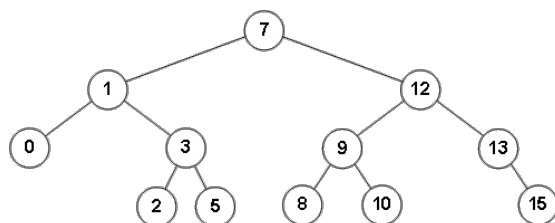
Bei der sequentiellen Suche wird der Datenbestand der Reihe nach durchsucht, bis der gegebene Suchwert gefunden ist. Der Algorithmus kann aber verbessert werden, indem der Datenbestand zuvor nach Zugriffshäufigkeit sortiert wird.

### 8.2 Binäre Suche

Grundvoraussetzung ist eine sortierte lineare Liste. Das Element in der Mitte der Liste wird gesucht. Wenn die gesuchte Zahl kleiner als das Mittenelement ist, wird links fortgesetzt, ansonsten rechts. Solange wird gesucht bis Element gefunden worden ist. Ansonsten kommt es zum Abbruch.

### 8.2 Binärer Suchbaum

Linker Nachfolger ist kleiner und rechter Nachfolger ist größer als das übergeordnete Element.



---

## 9. Graph

Ein Graph ist in der Graphentheorie eine abstrakte Struktur, die eine Menge von Objekten zusammen mit den zwischen diesen Objekten bestehenden Verbindungen repräsentiert.

Graphen unterscheiden sich in zwei Arten. Einen gerichteten bzw. ungerichteten Graphen.

Der Unterschied besteht darin, dass bei einem gerichteten Graphen die Beziehung (Kanten) zwischen 2 Elementen nicht umkehrbar ist.

Praktisches Anwendungsbeispiel: Routenplaner

### 9.1 Dijkstra

Er berechnet somit einen kürzesten Pfad zwischen dem gegebenen Startknoten und einem der (oder allen) übrigen Knoten in einem kantengewichteten Graphen (sofern dieser keine Negativkanten enthält).

Die Grundidee des Algorithmus ist es, immer derjenigen Kante zu folgen, die den kürzesten Streckenabschnitt vom Startknoten aus verspricht. Andere Kanten werden erst dann verfolgt, wenn alle kürzeren Streckenabschnitte beachtet wurden. Dieses Vorgehen gewährleistet, dass bei Erreichen eines Knotens kein kürzerer Pfad zu ihm existieren kann.

### 9.2 Warshall

Der Warshall Algorithmus dient dazu, den kürzesten Weg von allen Knoten zueinander zu finden. Die Idee dahinter ist es, nacheinander sämtliche Wege über die Zwischenknoten zu betrachten.

### 9.3 Graphdatenbank

Eine Graphdatenbank (oder graphenorientierte Datenbank) ist eine Datenbank, die Graphen benutzt, um stark vernetzte Informationen darzustellen und abzuspeichern. Graphdatenbanken bieten eine Reihe von spezialisierten Graphalgorithmen, um komplizierte Datenbankabfragen zu vereinfachen. So bieten sie beispielsweise Algorithmen, um Muster zu finden (Graph Pattern), Graphen zu traversieren, d. h. alle direkten und indirekten Nachbarn eines Knotens zu finden, kürzeste Pfade zwischen zwei Knoten zu berechnen, bekannte Graphstrukturen wie beispielsweise Cliquen zu finden oder Hotspots besonders stark vernetzter Regionen im Graph zu identifizieren.



---

## 9.4 Backtracking

Nach diesem Prinzip arbeitet das Backtracking-Verfahren.

- Man führt einen Algorithmus so lange aus, bis man an eine Grenze stößt.
- Ist das der Fall, kehrt man zum letzten Schritt zurück und testet einen anderen Folgeschritt.
- Versuche, eine gültige Teillösung auf dem Weg zum Ergebnis zu finden.
- Baue auf diese Teillösung den restlichen Weg zum Ziel auf.
- Ist das nicht möglich, versuche eine andere Teillösung zu finden.

Eine Teillösung kann man sich wie einen Knoten in einem Baum vorstellen. Jeder Knoten hat eine gewisse Anzahl von Ästen, die zu weiteren Knoten führen können. An jedem Ast hängt ein weiterer Baum, welcher mögliche Folgeschritte symbolisiert.

## 9.5 Adjazenzmatrix

Die Adjazenzmatrix ist dafür da, dass ein Graph auf in Tabellenform dargestellt werden kann.