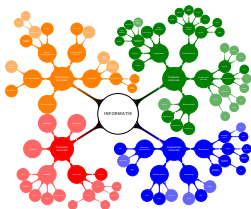


# Kapitel 31

## Algorithmendesign

Von gierigen Algorithmen und der Suche nach dem Minotaurus

Vorlesung Einführung in die Informatik 2 vom 15. April 2014 von Till Tantau



## Lernziele von Kapitel 31

1. Greedy-Verfahren kennen und implementieren können
2. Backtracking-Verfahren kennen und entwerfen können

# Gliederung von Kapitel 31

## 31.1 Was ist Algorithmen-Design?

## 31.2 Greedy-Verfahren

### 31.2.1 Idee

### 31.2.2 Beispiel: Münzproblem

### 31.2.3 Beispiel: Bin-Packing

### 31.2.4 Implementation

### 31.2.5 Vor- und Nachteile

## 31.3 Backtracking

### 31.3.1 Idee

### 31.3.2 Beispiel: Hamilton'sches Kreisproblem

### 31.3.3 Beispiel: Färbeproblem

### 31.3.4 Implementation

### 31.3.5 Vor- und Nachteile

Will man einen neuen Algorithmus entwerfen, so gibt es verschiedene Herangehensweisen:

- ▶ Wenn das zu lösende Problem einfach ist, so »sieht« man den Algorithmus vielleicht sofort.
- ▶ Man kennt schon einen Algorithmus, zumindest für ein sehr verwandtes Problem. Diesen wandelt man leicht ab.
- ▶ Man hat eine Eingebung/Erleuchtung.
- ▶ Man benutzt eines der *Standardverfahren des Algorithmendesign*.

31.1 Was ist  
Algorithmendesign? ◀

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches

Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

Eine Auswahl der Verfahren, die sich beim Design von Algorithmen bewährt haben:

1. Rekursion (Zurückführung des Problems auf ein vereinfachtes)
2. Teile-und-Herrsche (Aufteilen in Teilprobleme, Beispiele sind Mergesort und Quicksort)
3. Greedy-Verfahren (heute)
4. Backtracking (heute)
5. Rohe Gewalt (stures Durchprobieren aller Möglichkeiten, liebevoll Brute-Force genannt)
6. Dynamische Tabellen (intelligente Art der Rekursion)

All diese Verfahren *sind keine Algorithmen* sondern *Grundideen, nach denen man Algorithmen entwerfen kann.*

31.1 Was ist Algorithmendesign? ◀

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

► Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

Nimm, was du kriegen kannst!

# Die Grundidee von Greedy etwas wissenschaftlicher formuliert.

- ▶ Greedy-Verfahren dienen (in der Regel) zur Lösung von *Optimierungsproblemen*.
- ▶ Ausgehend von einer *leeren Teillösung* wird diese Teillösung *schrittweise vervollständigt*.
- ▶ Die durchgeführte Vervollständigung ist immer die, die *aktuell den meisten Vorteil bringt* (das Verfahren »giert« nach einem Vorteil).
- ▶ Wenn die Lösung fertig ist, terminiert das Verfahren.

▶ Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem  
Implementation

Vor- und Nachteile

## Das Optimierungsproblem

**Eingabe** Ein (Multi-)Menge an Münzen und ein Wert  $w$ .

**Ausgabe** Ein möglichst kleine Teilmenge der Münzen, deren Summe  $w$  ist.

## Beispiel

Beim Münzrückgabeproblem sollen 1,86 Euro zurückgegeben werden, folgende Münzen stehen zur Verfügung :



Dann lautet die richtige Ausgabe: dreimal 50 Cent, einmal 20 Cent, einmal 10 Cent, dreimal 2 Cent.

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren  
Idee

- Beispiel: Münzproblem
- Beispiel: Bin-Packing
- Implementation
- Vor- und Nachteile

31.3 Backtracking  
Idee  
Beispiel: Hamilton'sches  
Kreisproblem  
Beispiel: Färbeproblem  
Implementation  
Vor- und Nachteile



## Algorithmus

- ▶ Wir transferieren nach und nach Münzen vom *Eingabestapel* auf den *Rückgabestapel*.
- ▶ Anfangs ist der Rückgabestapel leer.
- ▶ Solange möglich, wiederhole Folgendes:
  1. Von den verbleibenden Münzen im Eingabestapel, identifiziere jede Münze, *die man noch auf den Rückgabestapel legen könnte, so dass dessen Wert höchstens  $w$  ist.*
  2. Unter all diesen Münzen, transferiere die Münze mit dem höchsten Wert vom Eingabestapel auf den Rückgabestapel.

31.1 Was ist  
Algorithmen-  
design?

31.2 Greedy-Verfahren

Idee

▶ Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

► Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

- Das Verfahren ist *sehr schnell und einfach*.
- Das Verfahren findet *aber nicht immer eine Lösung*.
- Die Euro-Münzen sind von ihren Werten her *so gemacht*, dass das Verfahren *immer funktioniert, wenn von jeder Münzart genügend Münzen vorhanden sind*.
- In diesem Fall liefert der Algorithmus sogar *immer eine optimale Lösung*.

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

► Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem  
Implementation

Vor- und Nachteile

## Das Bin-Packing-Problem

**Eingabe** Eine Liste von Objektgrößen  $(g_1, \dots, g_n)$  und eine Eimergröße  $b$ .

**Ausgabe** Zuordnung von Objekten zu möglichst wenigen Eimern, so dass die Summe der Größen aller Objekte, die demselben Eimer zugeordnet sind, maximal  $b$  ist.

31.1 Was ist  
Algorithmen-  
design?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

► Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

## Der First-Fit-Algorithmus

Für jeden Gegenstand tue folgendes:

1. Finde, von links beginnend, den ersten Eimer, in den der Gegenstand noch passt.
2. Platziere den Gegenstand in diesen Eimer.

# Wie implementiert man Greedy-Verfahren?

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

► Implementation

Vor- und Nachteile

### 31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

## Allgemeines Schema eines Greedy-Algorithmus

Solange die gefundene Teillösung noch nicht fertig ist, wiederhole:

- Unter allen möglichen nächsten Schritten, bestimme denjenigen, der *den meisten Gewinn bringt*.
- Erweitere die Teillösung entsprechend.

# Beispielcode für das Münzproblem

## Kapitel 31 Algorithmendesign

```
int[] wechselgeld (int[] geldbeutel, double betrag) {  
    // Münzwerte in Euros  
    double[] muenzwerte = {2, 1, .5, .2, .1, .05, .02, .01};  
    int[] wechselgeld = {0, 0, 0, 0, 0, 0, 0, 0};  
    while (betrag > 0) {  
        int beste_wahl = -1;  
        for (int i = 0; i < muenzwerte.length; i++) {  
            if (geldbeutel[i] > 0 && muenzwerte[i] <= betrag) {  
                beste_wahl = i;  
                break;  
            }  
        }  
        if (beste_wahl == -1) // Kein Wechselgeld mehr!  
            return null;  
        wechselgeld[beste_wahl]++;  
        geldbeutel[beste_wahl]--;  
        betrag = betrag - muenzwerte[best_wahl];  
    }  
    return wechselgeld;  
}
```

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee  
Beispiel: Münzproblem  
Beispiel: Bin-Packing  
► Implementation  
Vor- und Nachteile

31.3 Backtracking

Idee  
Beispiel: Hamilton'sches  
Kreisproblem  
Beispiel: Färbeproblem  
Implementation  
Vor- und Nachteile

31-14

# Zusammenfassung der Vor- und Nachteile von Greedy-Verfahren.

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

► Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

## Vorteile

- + Fast immer einsetzbar
- + Arbeiten immer sehr schnell
- + Produzieren in der Regel gute Lösungen
- + Leicht zu merken und zu programmieren

## Nachteile

- Liefern manchmal gar keine Lösungen
- Liefern manchmal nur schlechte Lösungen

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

► Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem  
Implementation

Vor- und Nachteile





# Warum wir in einem Labyrinth mit Greedy-Strategien nicht weiterkommen.

*Greedy* funktioniert *nicht*:

- ▶ In einem Labyrinth ist es *schwierig, lokal eine optimale Entscheidung* zu fällen.
- ▶ Kommt man an einer Kreuzung an, so erscheint jede Richtung »gleichermaßen gut oder schlecht«.

*Wie löst* man nun ein Labyrinthproblem?

- ▶ Man probiert die verschiedenen Richtungen nacheinander durch.
- ▶ Stößt man auf eine *Sackgasse*, so muss man die letzten Schritte rückwärts gehen (*backtracking*).
- ▶ Dazu führt man einen Ariadnefaden mit, den man aus- und einrollt.

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren  
Idee  
Beispiel: Münzproblem  
Beispiel: Bin-Packing  
Implementation  
Vor- und Nachteile

31.3 Backtracking  
▶ Idee  
Beispiel: Hamilton'sches  
Kreisproblem  
Beispiel: Färbeproblem  
Implementation  
Vor- und Nachteile

- ▶ Ein Algorithmus muss eine *Folge von Auswahlen* treffen.  
Beispiel: Welchen Knoten sollte ich als nächstes besuchen?  
Beispiel: In welchen Container sollte ich die Waren packen?
- ▶ Die *Strategie* ist, eine Folge von Auswahlen zu treffen, und diese, falls sie noch nicht optimal ist, *immer wieder in den letzten Schritten zu revidieren*.

31.1 Was ist  
Algorithmen-  
design?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

▶ Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem  
Implementation

Vor- und Nachteile

## Definition (Hamilton'scher Kreis)

Sei  $G$  ein Graph. Ein *Hamilton'scher Kreis* ist eine Folge von Knoten des Graphen, so dass

1. jeder Knoten genau einmal in der Folge vorkommt und
2. jeder Knoten mit seinem Nachfolger in der Folge in dem Graphen durch eine Kante verbunden ist und
3. der letzte mit dem ersten durch eine Kante verbunden ist.

## Definition (Hamilton'sches Kreisproblem)

Das *Hamilton'sche Kreisproblem* ist ein Entscheidungsproblem. Eingabe ist ein Graph. Die Frage ist, ob dieser einen Hamilton'schen Kreis enthält.

Offenbar ist das Hamilton'sche Kreisproblem etwas leichter als das allgemeine Handlungsreisendenproblem, da es eine »Vorstufe« dazu darstellt.

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren  
Idee  
Beispiel: Münzproblem  
Beispiel: Bin-Packing  
Implementation  
Vor- und Nachteile

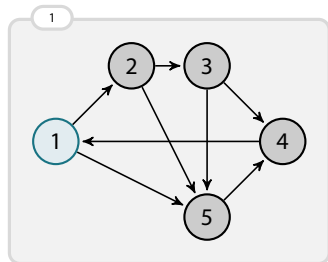
31.3 Backtracking  
Idee  
► Beispiel: Hamilton'sches  
Kreisproblem  
Beispiel: Färbeproblem  
Implementation  
Vor- und Nachteile

# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

## Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.



- 31.1 Was ist Algorithmendesign?
- 31.2 Greedy-Verfahren
  - Idee
  - Beispiel: Münzproblem
  - Beispiel: Bin-Packing
  - Implementation
  - Vor- und Nachteile
- 31.3 Backtracking
  - Idee
  - Beispiel: Hamilton'sches Kreisproblem
  - Beispiel: Färbeproblem
  - Implementation
  - Vor- und Nachteile

# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

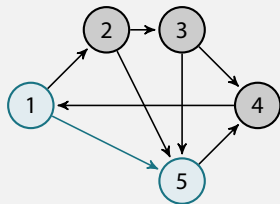
## Kapitel 31 Algorithmendesign

### Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.

2



31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

► Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

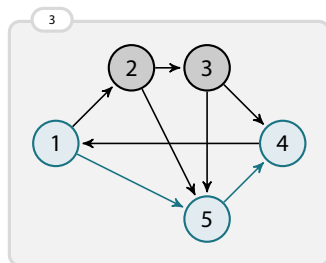
Vor- und Nachteile

# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

## Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.



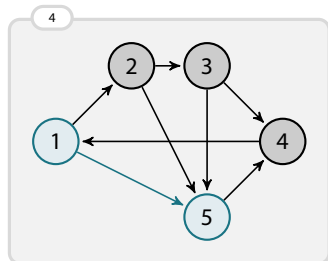
- 31.1 Was ist Algorithmendesign?
- 31.2 Greedy-Verfahren
  - Idee
  - Beispiel: Münzproblem
  - Beispiel: Bin-Packing
  - Implementation
  - Vor- und Nachteile
- 31.3 Backtracking
  - Idee
  - Beispiel: Hamilton'sches Kreisproblem
  - Beispiel: Färbeproblem
  - Implementation
  - Vor- und Nachteile

# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

## Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.



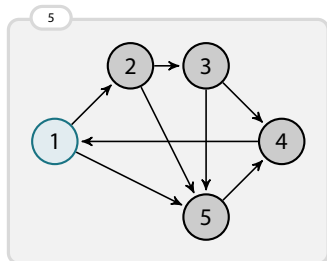
- 31.1 Was ist Algorithmendesign?
- 31.2 Greedy-Verfahren
  - Idee
  - Beispiel: Münzproblem
  - Beispiel: Bin-Packing
  - Implementation
  - Vor- und Nachteile
- 31.3 Backtracking
  - Idee
  - Beispiel: Hamilton'sches Kreisproblem
  - Beispiel: Färbeproblem
  - Implementation
  - Vor- und Nachteile

# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

## Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.



- 31.1 Was ist Algorithmendesign?
- 31.2 Greedy-Verfahren
  - Idee
  - Beispiel: Münzproblem
  - Beispiel: Bin-Packing
  - Implementation
  - Vor- und Nachteile
- 31.3 Backtracking
  - Idee
  - Beispiel: Hamilton'sches Kreisproblem
  - Beispiel: Färbeproblem
  - Implementation
  - Vor- und Nachteile



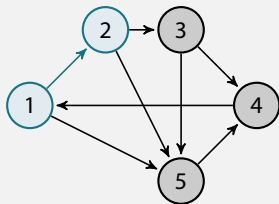
# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

## Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.

6



31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

► Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

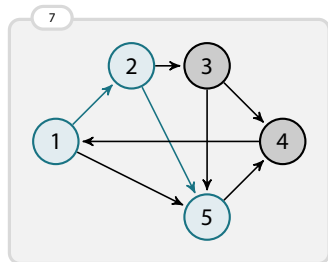
Vor- und Nachteile

# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

## Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.



- 31.1 Was ist Algorithmendesign?
- 31.2 Greedy-Verfahren
  - Idee
  - Beispiel: Münzproblem
  - Beispiel: Bin-Packing
  - Implementation
  - Vor- und Nachteile
- 31.3 Backtracking
  - Idee
  - Beispiel: Hamilton'sches Kreisproblem
  - Beispiel: Färbeproblem
  - Implementation
  - Vor- und Nachteile

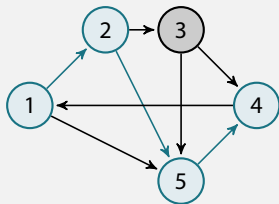
# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

## Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.

8



- 31.1 Was ist Algorithmendesign?
- 31.2 Greedy-Verfahren
  - Idee
  - Beispiel: Münzproblem
  - Beispiel: Bin-Packing
  - Implementation
  - Vor- und Nachteile
- 31.3 Backtracking
  - Idee
  - Beispiel: Hamilton'sches Kreisproblem
  - Beispiel: Färbeproblem
  - Implementation
  - Vor- und Nachteile

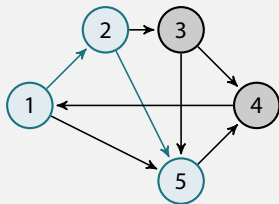
# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

## Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.

9



31.1 Was ist Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

► Beispiel: Hamilton'sches Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

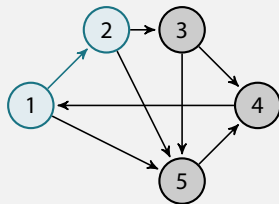
## Kapitel 31 Algorithmendesign

### Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.

10



31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

► Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

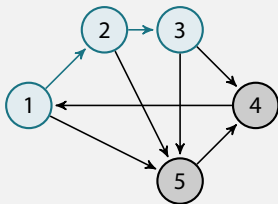
# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

## Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.

11



31.1 Was ist Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

► Beispiel: Hamilton'sches Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

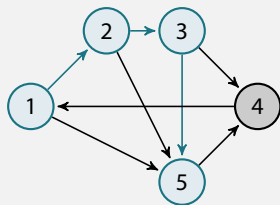
# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

## Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.

12



31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

► Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

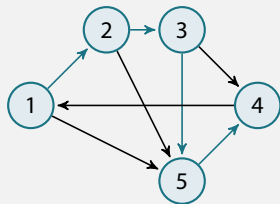
# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

## Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.

13



31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

► Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile



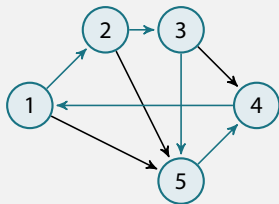
# Backtracking-Algorithmus zum Finden Hamilton'scher Kreise

## Algorithmus

Wiederhole Folgendes, bis es nicht mehr geht:

1. Erweitere den Pfad beliebig, bis es nicht mehr geht.
2. Wenn alle Knoten besucht sind und es eine Kante vom Ende zum Anfang gibt, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis der Pfad anders fortgesetzt werden kann.

14



31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

► Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

## Die Problemstellung

**Eingabe** Ein Graph.

**Lösungen** Färbung des Graphen (je zwei durch eine Kante verbundenen Knoten müssen eine unterschiedliche Farbe haben)

**Maß** Anzahl der benutzten Farben

**Ziel** Minimierung

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren  
Idee  
Beispiel: Münzproblem  
Beispiel: Bin-Packing  
Implementation  
Vor- und Nachteile

31.3 Backtracking  
Idee  
Beispiel: Hamilton'sches  
Kreisproblem  
► Beispiel: Färbeproblem  
Implementation  
Vor- und Nachteile

## Algorithmus zur 3-Färbung

31-21

Wiederhole Folgendes, bis es nicht mehr geht:

1. Färbe die Knoten mit drei Farben, bis es nicht mehr geht.
2. Wenn alle Knoten gefärbt sind, höre auf.
3. Sonst mache so viele der letzten Schritte rückgängig, bis man mit einer andere Färbung weitermachen kann.

# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

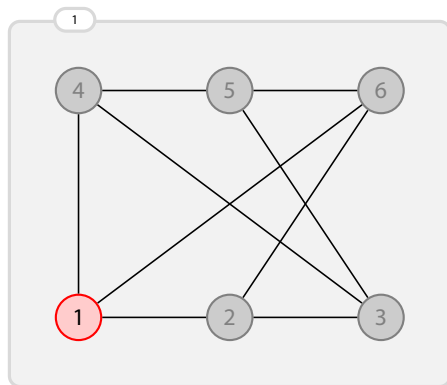
Vor- und Nachteile

### 31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

- Beispiel: Färbeproblem
- Implementation
- Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

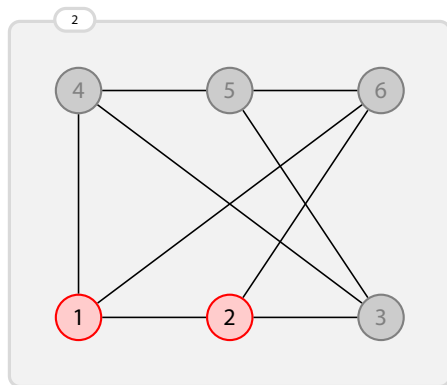
Vor- und Nachteile

### 31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

- Beispiel: Färbeproblem
- Implementation
- Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

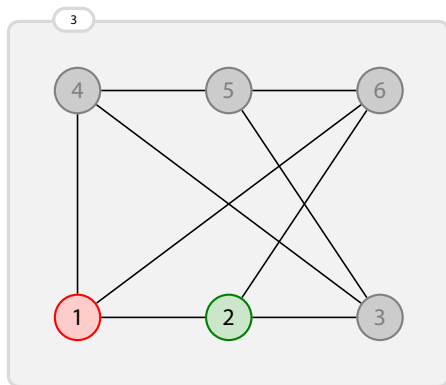
Vor- und Nachteile

### 31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

- Beispiel: Färbeproblem
- Implementation
- Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

### 31.3 Backtracking

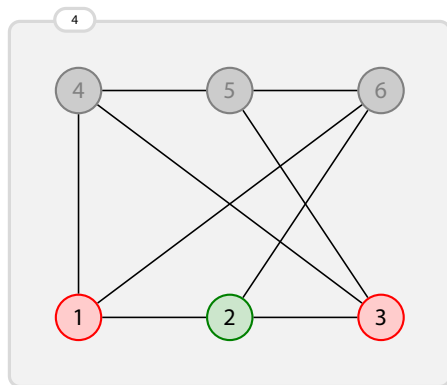
Idee

Beispiel: Hamilton'sches  
Kreisproblem

► Beispiel: Färbeproblem

Implementation

Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

### 31.3 Backtracking

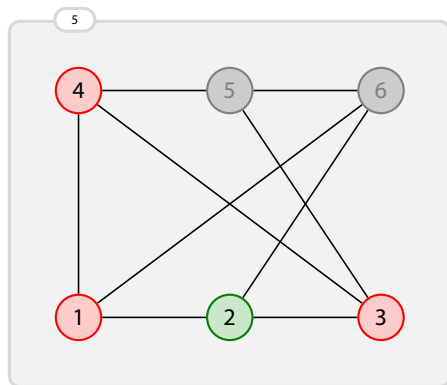
Idee

Beispiel: Hamilton'sches  
Kreisproblem

► Beispiel: Färbeproblem

Implementation

Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

### 31.3 Backtracking

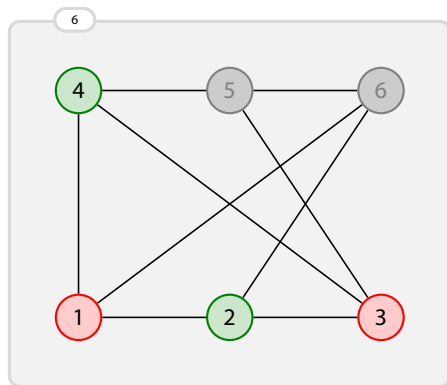
Idee

Beispiel: Hamilton'sches  
Kreisproblem

► Beispiel: Färbeproblem

Implementation

Vor- und Nachteile





# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

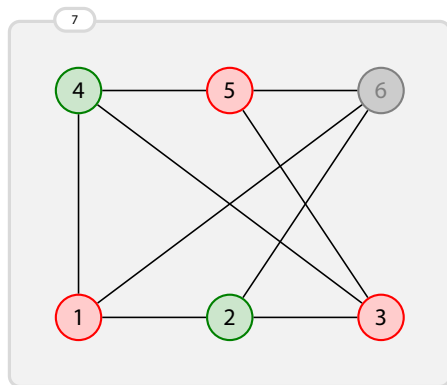
Vor- und Nachteile

### 31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

- Beispiel: Färbeproblem
- Implementation
- Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

### 31.3 Backtracking

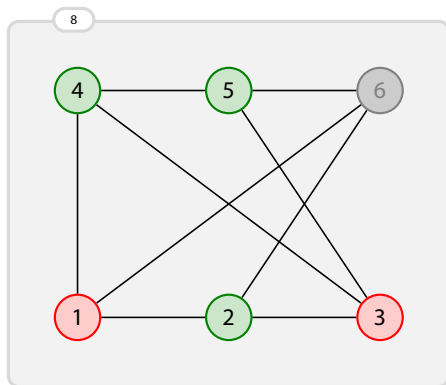
Idee

Beispiel: Hamilton'sches  
Kreisproblem

► Beispiel: Färbeproblem

Implementation

Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

### 31.3 Backtracking

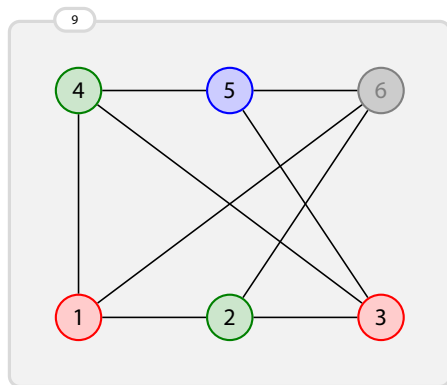
Idee

Beispiel: Hamilton'sches  
Kreisproblem

► Beispiel: Färbeproblem

Implementation

Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

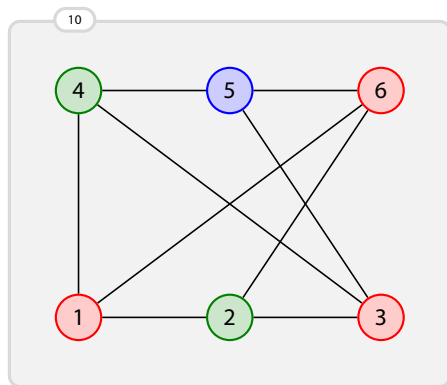
Vor- und Nachteile

### 31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

- Beispiel: Färbeproblem
- Implementation
- Vor- und Nachteile



31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

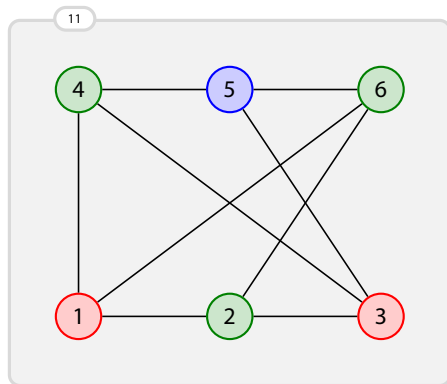
Idee

Beispiel: Hamilton'sches  
Kreisproblem

► Beispiel: Färbeproblem

Implementation

Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

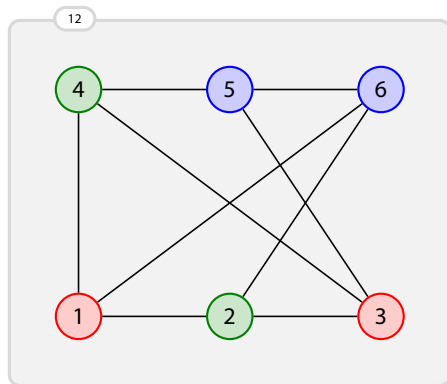
Vor- und Nachteile

### 31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

- Beispiel: Färbeproblem
- Implementation
- Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

### 31.3 Backtracking

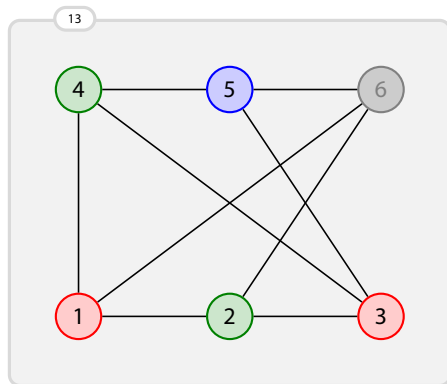
Idee

Beispiel: Hamilton'sches  
Kreisproblem

► Beispiel: Färbeproblem

Implementation

Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

### 31.3 Backtracking

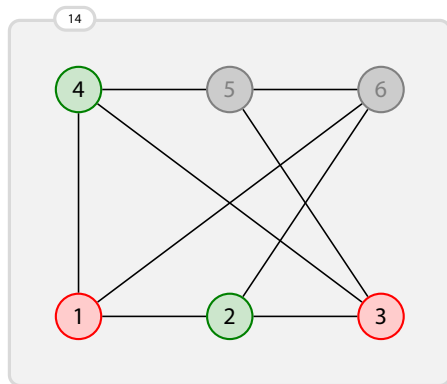
Idee

Beispiel: Hamilton'sches  
Kreisproblem

► Beispiel: Färbeproblem

Implementation

Vor- und Nachteile





# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

### 31.3 Backtracking

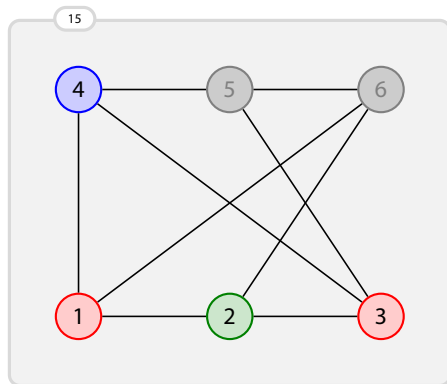
Idee

Beispiel: Hamilton'sches  
Kreisproblem

► Beispiel: Färbeproblem

Implementation

Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

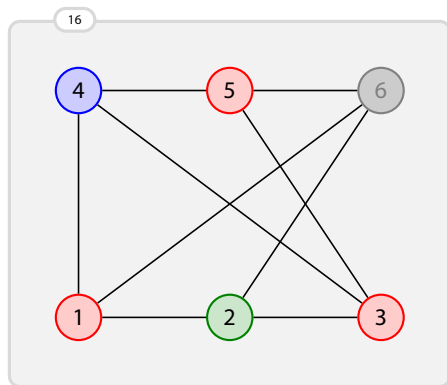
Vor- und Nachteile

### 31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

- Beispiel: Färbeproblem
- Implementation
- Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

### 31.3 Backtracking

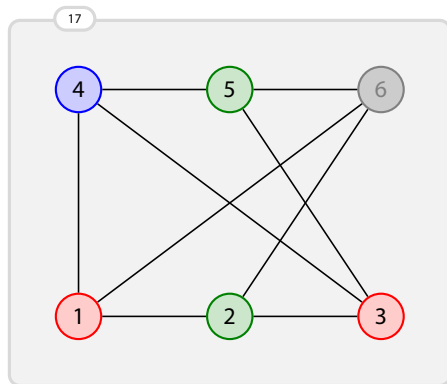
Idee

Beispiel: Hamilton'sches  
Kreisproblem

► Beispiel: Färbeproblem

Implementation

Vor- und Nachteile



# Färben eines Graphen mittels Backtracking

## Kapitel 31 Algorithmendesign

### 31.1 Was ist Algorithmendesign?

### 31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

### 31.3 Backtracking

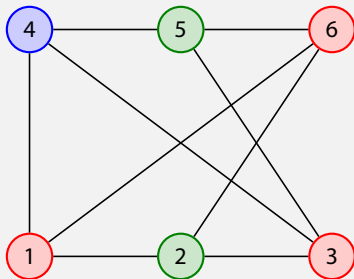
Idee

Beispiel: Hamilton'sches  
Kreisproblem

- Beispiel: Färbeproblem
- Implementation
- Vor- und Nachteile

31-22

18



31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

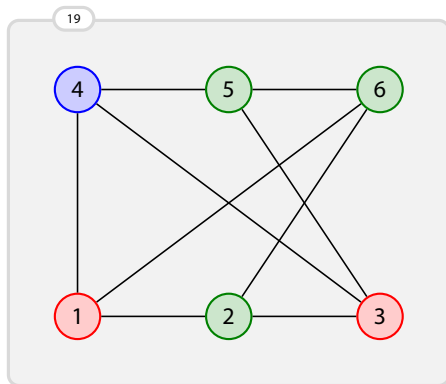
Beispiel: Hamilton'sches  
Kreisproblem

► Beispiel: Färbeproblem

Implementation

Vor- und Nachteile

31-22



31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

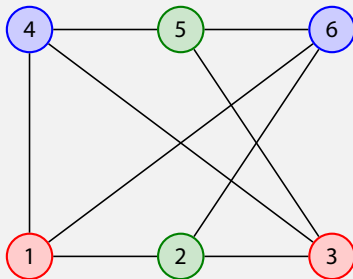
Idee

Beispiel: Hamilton'sches  
Kreisproblem

- Beispiel: Färbeproblem
- Implementation  
Vor- und Nachteile

31-22

20



# Wie implementiert man Backtracking?

Es gibt zwei Ansätze zur Implementation von Backtracking-Verfahren:

1. Beim *iterativen Ansatz* wird der »Ariadnefaden« in einem Array gespeichert. Jede Array-Position gibt eine der bisherigen Entscheidungen an.

Immer, wenn ein Backtracking nötig ist, wird die letzten Entscheidungen geändert. Lässt sich diese nicht mehr ändern, so wird zur vorletzten zurückgegangen und diese geändert. Lässt sich auch diese nicht mehr ändern, so geht es zurück zur vorvorletzten und so weiter.

2. Beim *rekursiven Ansatz* wird der »Ariadnefaden« indirekt durch den Compiler verwaltet.

Jedesmal, wenn eine Entscheidung nötig ist, ruft sich eine Methode rekursiv für alle möglichen Entscheidungen auf.

Insgesamt ist ein Backtracking in der Regel schwierig zu implementieren.

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

► Implementation

Vor- und Nachteile

# Zusammenfassung der Vor- und Nachteile von Backtracking-Verfahren.

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

► Vor- und Nachteile

## Vorteile

- + Liefern immer optimale Ergebnisse.
- + Funktionieren im Notfall immer.

## Nachteile

- Schwierig zu programmieren.
- Funktionieren nur bei sehr kleinen Eingaben (max. 30 Knoten).
- Werden niemals (!) bei wirklich großen Graphen (über 500 Knoten) funktionieren.



## Grundlegende Verfahren des Algorithmendesigns

- ▶ Rekursion
- ▶ Teile-und-Herrsche
- ▶ Greedy-Verfahren
- ▶ Backtracking
- ▶ Brute-Force
- ▶ Dynamische Tabellen

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem  
Implementation

▶ Vor- und Nachteile

31.1 Was ist  
Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem

Implementation

► Vor- und Nachteile

## Greedy-Verfahren

- Idee: Erweitere eine Teillösung immer in lokal optimaler Weise.
- Vorteile: Schnell, einfach, oft gute Lösungen
- Nachteile: Führt nicht immer zum Ziel.
- Anwendungsbeispiele: Münzrückgabe, Bin-Packing

## Backtracking-Verfahren

- ▶ Idee: Erweitere eine Teillösung in allen lokal möglichen Weisen und revidiere getroffene Entscheidungen, wenn keine Lösung gefunden wurde.
- ▶ Vorteile: Findet immer die optimale Lösung.
- ▶ Nachteile: Dauert schon bei kleinen Eingabegrößen sehr oder zu lange.
- ▶ Anwendungsbeispiele: Münzrückgabe, Bin-Packing, Färbung, Hamilton'sches Kreisproblem

31.1 Was ist Algorithmendesign?

31.2 Greedy-Verfahren

Idee

Beispiel: Münzproblem

Beispiel: Bin-Packing

Implementation

Vor- und Nachteile

31.3 Backtracking

Idee

Beispiel: Hamilton'sches  
Kreisproblem

Beispiel: Färbeproblem  
Implementation

▶ Vor- und Nachteile