



Informatik I WS 07/08

Tutorium 24

31.01.08

Bastian Molkenthin

E-Mail: infotut@sunshine2k.de

Web: <http://infotut.sunshine2k.de>



Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825



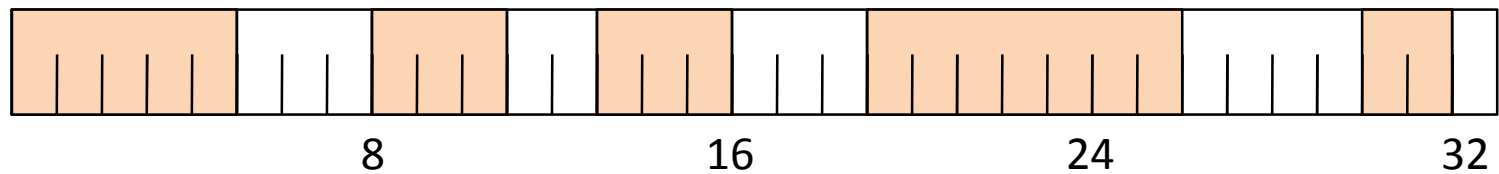
- Anmeldung Hauptklausur : allerspätestens bis **17.2.08** (*besser früher als später*)
Informationen dazu unter 'Aktuelles' auf der Info1-Seite!
- Für den *Erhalt des Übungsschein* muss man sich ebenfalls anmelden! Wichtig!
Infos dazu auch unter 'Aktuelles' auf der Info1-Seite!
Auch anmelden wenn der Schein noch nicht ganz sicher ist.
- Minimale Punkte für Scheinerwerb: **273** Theoriepunkte *und* **96** Praxispunkte.
- Am nächsten Montag (4.2.08) findet keine Rechnerübung statt !!

Falls wider Erwarten die restlichen Rechnerübungen nicht ausreichen, können wir eine zusätzliche vereinbaren...



- Wozu?

➔ Freier und belegter Speicher muss repräsentiert werden

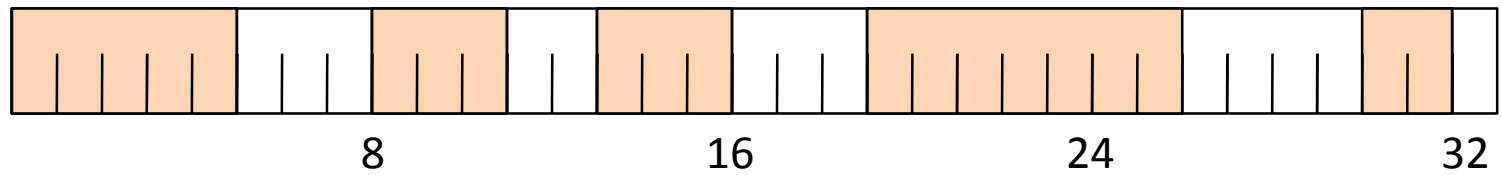


- 1. Möglichkeit: Bitmap

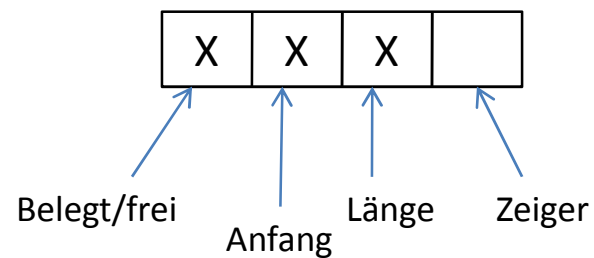
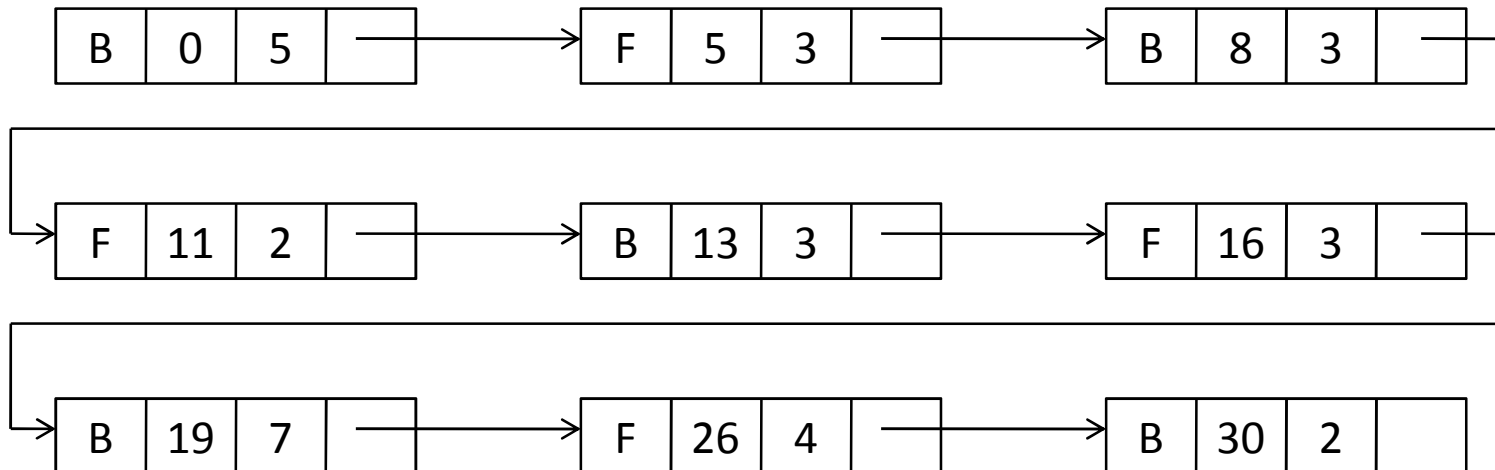
- Feste Breite wählen (hier 8 Bit) und in einem Feld die Speicherbelegung binär kodieren (1 = belegt, 0 = frei)

```
11111000  
11100111  
00011111  
11000011
```

Speicherverwaltung



- 2. Möglichkeit: Verkettete Liste

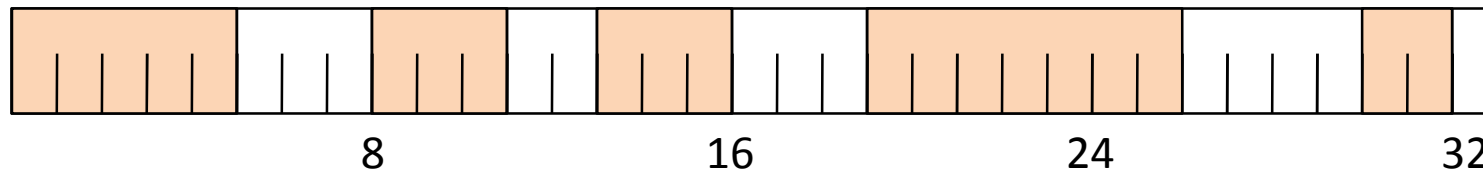




- Strategien für die Belegung von Speicher:
 - First Fit
 - Die erste passende Lücke wird verwendet
 - Best Fit
 - Die kleinste passende Lücke wird verwendet
 - Worst Fit
 - Die größte passende Lücke wird verwendet

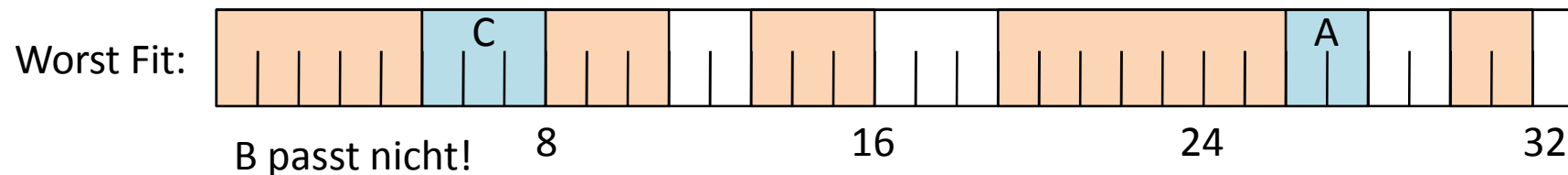
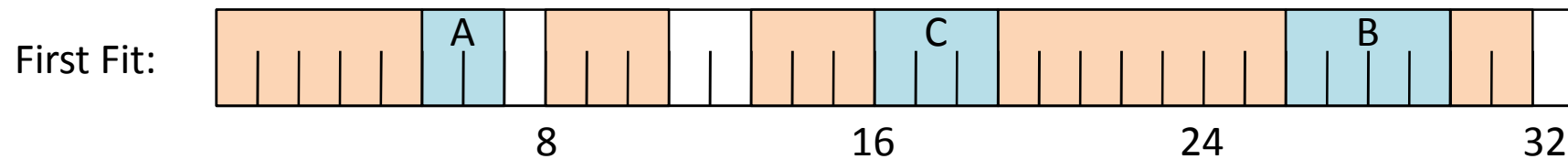
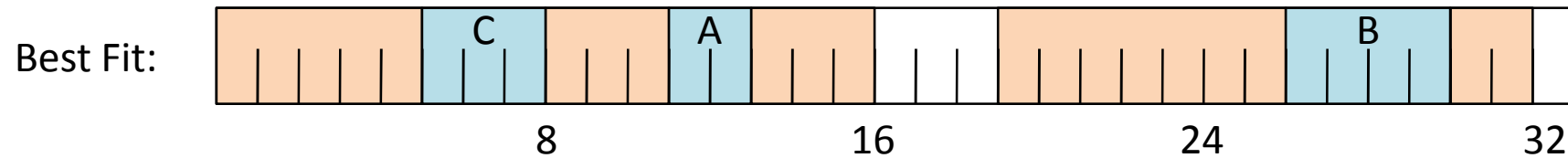
- Probleme:
 - Speicherverschnitt
 - zu kleine Lücken

Speicherverwaltung - Beispiel



Folgende Speicherbereiche werden der Reihe nach alloziert:

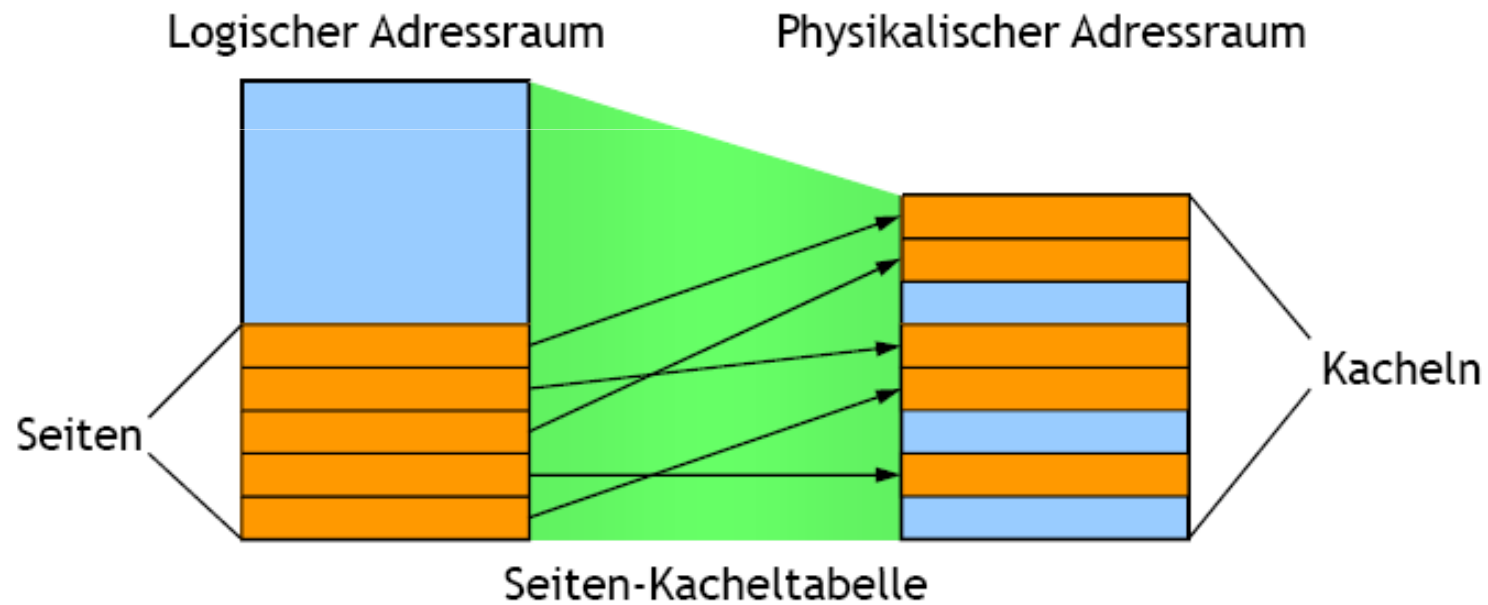
Speicherbereich	Größe
A	2
B	4
C	3





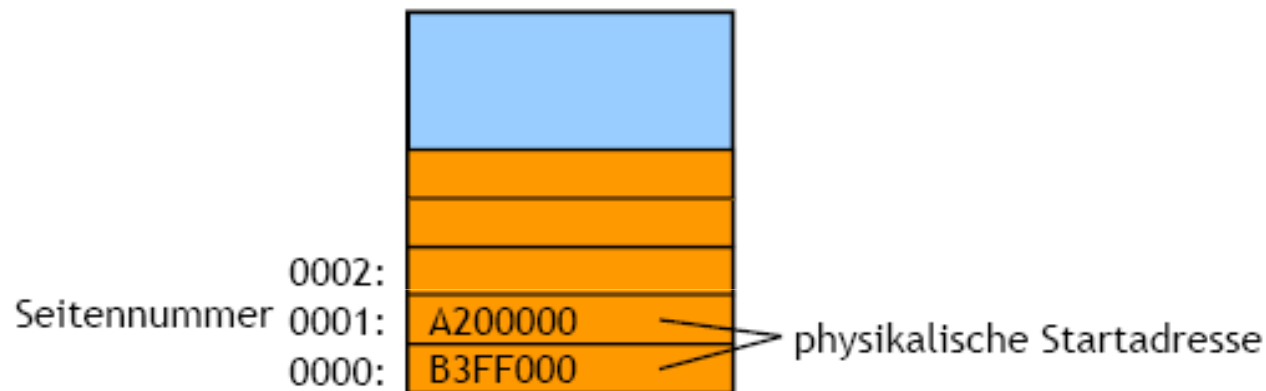
Wofür Paging?

- verhindert externe Fragmentierung
- Mehr Speicher adressierbar, als physikalischer Speicher verfügbar





- Der Speicher wird in Seiten der Größe n Byte eingeteilt
- Diese Einteilung wird in einer Kacheltabelle gespeichert



- Jeder Eintrag besteht aus einer Seitennummer, und einer physikalischen Speicheradresse, bei der diese Seite beginnt.
- Die logische Adresse besteht dann aus der Seitennummer und $(\log n)$ Bytes, die die Adresse innerhalb der Seite angeben.

Paging - Aufgabe



Gegeben sei folgender Ausschnitt aus einer Seiten-Kacheltabelle:

Seitennummer	Startadresse
01	0000
02	4A00
03	FF00

Die Seitengröße im System betrage 256 Byte und der Speicher sei Byteweise adressierbar.
Die Adresslänge betrage 16 Bit.

- Wie viele Bits werden benötigt, um innerhalb einer Seite ein Byte zu adressieren?

➔ $\log_2(256) = 8 \text{ Bit}$

- Aus wievielen Bits besteht folglich eine Seitennummer?

➔ $16 - 8 \text{ Bit} = 8 \text{ Bit}$

Paging - Aufgabe






Gegeben sei folgender Ausschnitt aus einer Seiten-Kacheltabelle:

Seitennummer	Startadresse
01	0000
02	4A00
03	FF00

Die Seitengröße im System betrage 256 Byte und der Speicher sei Byteweise adressierbar.
Die Adresslänge betrage 16 Bit.

- Wandeln sie die folgenden logischen Adressen in Physische um:

0123  0023
03FF  FFFF
02A7  4AA7

Paging - Aufgabe






Gegeben sei folgender Ausschnitt aus einer Seiten-Kacheltabelle:

Seitennummer	Startadresse
01	0000
02	4A00
03	FF00

Die Seitengröße im System betrage 256 Byte und der Speicher sei Byteweise adressierbar.
Die Adresslänge betrage 16 Bit.

- Wandeln sie die folgenden physischen Adressen in Logische um:


00EE  01EE
4A47  0247
FF00  0300

Java - Bedingungsoperator



Der Bedingungsoperator erlaubt es, den Wert eines Ausdrucks von einer Bedingung Abhängig zu machen, ohne dass dazu eine if-Anweisung verwendet werden muss. Die Operanden sind durch `?` beziehungsweise `:` voneinander getrennt.

```
if (a > b) then
    max = a;
else
    max = b;
```



```
max = (a > b) ? a : b;
```

Gegeben sei eine int-Variable `y`. Formuliere einen Ausdruck, der einer Variablen `x` den Betrag von `y` zuweist.

Lösung: `x = (y >= 0) ? y : -y;`

Kann auch geschachtelt sein – was macht folgende Funktion:

```
int sign( int n ) {
    return (n < 0) ? -1 : ((n > 0) ? 1 : 0);
}
```

 Liefert abhängig vom Vorzeichen von `n` -1, 0 oder 1.



- Ähnlich den Semi-Thue-Systemen, wieder zwei Grundmengen:
 - Zeichenvorrat Σ
 - Menge von Regeln T
- Größter Unterschied ist die definierte Reihenfolge bei der Regelanwendung:
 - Wende stets die erste mögliche Regel an (Nummerieren!)
 - Wende sie so weit links wie möglich an
 - Nach eine Halteregel (.) beende den Algorithmus
 - Wenn keine Regel mehr anwendbar ist, beende den Algorithmus

➔ Markov-System ist deterministisch!

Markov - Algorithmen (2)



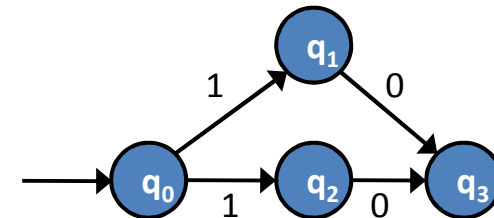
Ein Textersetzungssystem heißt **deterministisch** , wenn

wenn zu jeder Zeit immer nur eine Regel auf genau ein Teilwort anwendbar ist. Nichtdeterministisch bedeutet, dass für eine Eingabe das Verhalten des Systems unterschiedliche sein kann.

Ein Textersetzungssystem heißt **determiniert** , wenn

dieselbe Eingabe immer dieselbe Ausgabe ergibt. Nichtdeterminiert bedeutet, dass für eine Eingabe die Ausgabe des Systems unterschiedlich sein kann.

- Frage: nichtdeterminiert \Rightarrow nichtdeterministisch ✓
nichtdeterministisch \Rightarrow nichtdeterminiert ✗
deterministisch \Rightarrow determiniert ✓
determiniert \Rightarrow deterministisch ✗



- Wichtiger "Trick" bei Markov-Algorithmen:
 - Oft ist es nützlich, die Bearbeitungsposition zu speichern
➡ Verwendung sogenannter Schiffchen

Aufgabe Markov-Algo



Um das Zweierkomplement einer Binärzahl zu erhalten, muss diese zunächst invertiert und anschließend um 1 inkrementiert werden. Entwerfen Sie einen Markov-Algorithmus, welcher das Zweierkomplement einer gegebenen Binärzahl berechnet. Beispiel: $101 \Rightarrow 010 \Rightarrow 011$

$$\Sigma = \{ a, b, \alpha, \beta \}$$

$$T = \left\{ \begin{array}{l} (1) \quad \alpha 0 \rightarrow 1\alpha, \\ (2) \quad \alpha 1 \rightarrow 0\alpha, \\ (3) \quad 0\alpha \rightarrow .1, \\ (4) \quad 1\alpha \rightarrow \beta 0, \\ (5) \quad 1\beta \rightarrow \beta 0, \\ (6) \quad 0\beta \rightarrow .1, \\ (7) \quad \beta \rightarrow .1, \\ (8) \quad \varepsilon \rightarrow \alpha \quad \end{array} \right\}$$

Diese Aufgabe ist der inoffiziellen Informatik-I Probeklausur 2008 der Universität Karlsruhe entnommen.
Die Aufgabe wurde von Matthias Bartelt erstellt.

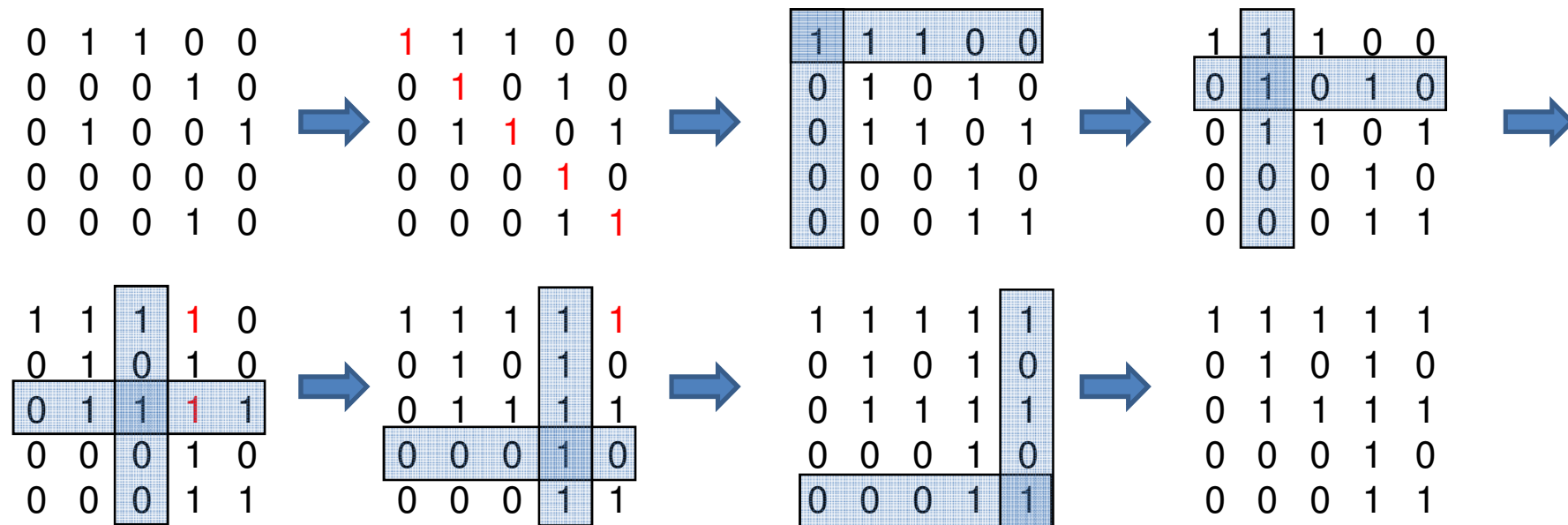
Warshall Algorithmus



Algorithmus zur Bestimmung der reflexiven transitiven Hülle

Gegeben sei ein Graph als Adjazenzmatrix - anschauliche Durchführung:

- Ergänze sämtliche Knoten um eine Schlinge (Einsen auf der Diagonalen der Matrix)
- Betrachte k-te Zeile der Matrix. Alle weiteren Zeilen, welche in **Spalte** k ebenfalls eine 1 beinhalten können verändert werden:
 - Sämtliche Nullen in diesen Zeilen werden durch eine Eins ersetzt, sofern die Ausgangszeile k an dieser Stelle eine 1 enthält





Fragen ???



Viel Spaß mit dem Übungsblatt!