MATHEMATISCHE ANALYSE VON ALGORITHMEN

Michael Drmota

Institut für Diskrete Mathematik und Geometrie,

TU Wien

michael.drmota@tuwien.ac.at

www.dmg.tuwien.ac.at/drmota/

Ringvorlesung SS 2008, TU Wien

Algorithmus

- "Kochrezept"
- Rechenoperationen
- Datenstrukturen
- Graphen
- **Aufwand = ?**

Inhalt

- Mergesort
- Karatsuba-Multiplikation
- Master-Theorem
- Euklidischer Algorithmus
- Quicksort – Binäre Suchbäume
- Travelling Salesman-Problem

Mergesort

Sortieren von Daten

Datensatz: $4, 7, 2, 6, 1, 3, 10, 8, 5, 9$

1. Aufteilen in 2 gleich große Teil-Datensätze

   $4, 7, 2, 6, 1$ und $3, 10, 8, 5, 9$

2. Sortieren der Teil-Datensätze

   $1, 2, 4, 6, 7$ und $3, 5, 8, 9, 10$

3. **Mergen** der Teil-Datensätze

   $1\ 2\ 4\ 6\ 7$
   $3\ 5\ 8\ 9\ 10$ $\longrightarrow$ $1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10$

Mergesort

$T(n)$ ... Aufwand des Sortierens von n Datensätzen

$$\boxed{T(n) = 2\,T(n/2) + O(n)}$$

$$\Longrightarrow \quad \boxed{\boxed{T(n) = O(n \log n)}}$$

Das ist deutlich schneller als das *naive Sortieren*, dessen Aufwand $O(n^2)$ ist.

Karatsuba-Multiplikation

Binärsystem

$$13 = 2^3 + 2^2 + 2^0 = (1101)_2$$

Aufwand beim üblichen Multiplizieren zweier n-stelliger Zahlen x, y:

$$O(n^2)$$

Karatsuba-Multiplikation

x, y ... n-stellige Zahlen (im Binärsystem)

1. Aufteilen in zwei gleich große Blöcke x_1, x_2 bzw. y_1, y_2 (mit $m = n/2$ Stellen):

$$x = 2^m x_1 + x_2, \qquad y = 2^m y_1 + y_2.$$

2. Bestimmen von

$$X = x_1 y_1, \quad Y = x_2 y_2, \quad \boxed{Z = x_1 y_2 + x_2 y_1 = (x_1 + x_2)(y_1 + y_2) - X - Y}$$

3. Zusammenfassen

$$\begin{aligned} xy &= (2^m x_1 + x_2)(2^m y_1 + y_2) = 2^{2m} x_1 y_1 + 2^m (x_1 y_2 + x_2 y_1) + x_2 y_2 \\ &= 2^{2m} X + 2^m Z + Y. \end{aligned}$$

Karatsuba-Multiplikation

$T(n)$ ... Aufwand des Multiplizierens von zwei n-stelligen Zahlen

$$\boxed{T(n) = 3\,T(n/2) + O(n)}$$

$$\Longrightarrow \quad \boxed{\boxed{T(n) = O(n^{\alpha})}}, \quad \alpha = \frac{\log 3}{\log 2} = 1.585...$$

Das ist deutlich schneller als das *gewöhnliche Multiplizieren*.

Karatsuba-Multiplikation

Die Idee zu dieser Vereinfachung stammt von A. Karatsuba aus dem Jahr 1960.

In der Zwischenzeit gibt es noch viel effizientere Algorithmen, ewta den Schönhage-Strassen-Algorithmus, dessen Aufwand die Größenordnung $O(n \log n)$ hat.

Master-Theorem

Divide and Conquer-Algorithmen

$$T(n) = aT(n/b) + O(n^c)$$

$\alpha := \frac{\log a}{\log b}$

$$\Longrightarrow \quad T(n) = \begin{cases} O(n^\alpha) & \text{für } \alpha > c, \\ O(n^\alpha \log n) & \text{für } \alpha = c, \text{ und} \\ O(n^c) & \text{für } \alpha < c. \end{cases}$$

Euklidischer Algorithmus

a, b ... ganze Zahlen

$d = \mathrm{ggT}(a, b)$ ... größter gemeinsamer Teiler (ggT)

$$\boxed{\mathrm{ggT}(a, b) = \mathrm{ggT}(a - b, b)}$$

Anwendung:

$$\mathrm{ggT}(59, 11) = \mathrm{ggT}(48, 11) = \cdots = \mathrm{ggT}(15, 11) = \mathrm{ggT}(4, 11),$$

$$\mathrm{ggT}(4, 11) = \mathrm{ggT}(4, 7) = \mathrm{ggT}(4, 3) = \mathrm{ggT}(1, 3) = 1,$$

$$\Longrightarrow \quad \boxed{\mathrm{ggT}(59, 11) = 1}.$$

Euklidischer Algorithmus

Fortgesetzte Division mit Rest:

$$\begin{aligned} 59 &= 5 \cdot 11 + 4, \\ 11 &= 2 \cdot 4 + 3, \\ 4 &= 1 \cdot 3 + 1. \end{aligned}$$

Euklidischer Algorithmus

Fortgesetzte Division mit Rest:

$$
\begin{aligned}
59 &= 5 \cdot 11 + 4, \\
11 &= 2 \cdot 4 + 3, \\
4 &= 1 \cdot 3 + 1, \\
3 &= 3 \cdot 1 + 0.
\end{aligned}
$$

Euklidischer Algorithmus

Fortgesetzte Division mit Rest:

$$
\begin{aligned}
59 &= 5 \cdot 11 + 4, \\
11 &= 2 \cdot 4 + 3, \\
4 &= 1 \cdot 3 + 1, \\
3 &= 3 \cdot 1 + 0.
\end{aligned}
$$

$$
\begin{aligned}
a &= q_1 b + r_1, \\
b &= q_2 r_1 + r_2, \\
r_1 &= q_3 r_2 + r_3 \\
&\vdots \\
r_k &= q_{k+2} r_{k+1} + r_{k+2} \\
r_{k+1} &= q_{k+3} r_{k+2} + 0
\end{aligned}
$$

Euklidischer Algorithmus

Fortgesetzte Division mit Rest:

$$
\begin{aligned}
59 &= 5 \cdot 11 + \boxed{4}, \\
11 &= 2 \cdot 4 + \boxed{3}, \\
4 &= 1 \cdot 3 + \boxed{1}, \\
3 &= 3 \cdot 1 + \boxed{0}.
\end{aligned}
$$

$$
\begin{aligned}
a &= q_1 b + \boxed{r_1}, \\
b &= q_2 r_1 + \boxed{r_2}, \\
r_1 &= q_3 r_2 + \boxed{r_3} \\
&\vdots \\
r_k &= q_{k+2} r_{k+1} + \boxed{r_{k+2}} \\
r_{k+1} &= q_{k+3} r_{k+2} + \boxed{0}
\end{aligned}
$$

Euklidischer Algorithmus

Aufwand

Die Anzahl $k+2$ der Divisionen gibt den Aufwand an:

$$b > r_1 > r_2 > \cdots > r_k > r_{k+1} > r_{k+2} > 0$$

und $\boxed{r_{k+2} = \mathsf{ggT}(a,b)}$.

$$r_j = q_{j+2} r_{j+1} + r_{j+2} \geq r_{j+1} + r_{j+2} > 2 r_{j+2}$$

$$\Longrightarrow \quad \boxed{r_{j+2} \leq \frac{1}{2} r_j}$$

$$\Longrightarrow \quad \boxed{\boxed{\text{Aufwand} = O(\log b)}}.$$

Euklidischer Algorithmus

Mittlerer Aufwand

Man betrachtet alle ganze Zahlen (a, b) mit $1 \leq b < a \leq n$.

Dann ist die durchschnittliche Anzahl der Divisionsschritte

$$\frac{12 \log 2}{\pi^2} \log n$$

und die mittlere Abweichung von diesem Wert

$$O(\sqrt{\log n}).$$

Weiters gilt ein zentraler Grenzwertsatz. (Die Verteilung um den Mittelwert wird asymptotisch durch die Gaußsche Normalverteilung beschrieben).

Quicksort

Datensatz: $4, 6, 3, 5, 1, 8, 2, 7$

1. Bestimmen eines Pivotelements (z.B. 4) und Aufteilen in 2 Teil-Datensätze, in die Elemente, die kleiner als das Pivotelement sind und jene, die größer sind:

   $$3, 1, 2 \qquad 4 \qquad 6, 5, 8, 7$$

2. Sortieren der Teil-Datensätze:

   $$1, 2, 3 \qquad \text{und} \qquad 5, 6, 7, 8$$

3. Zusammensetzen der Teil-Datensätze und des Pivotelments:

   $$1, 2, 3, 4, 5, 6, 7, 8$$

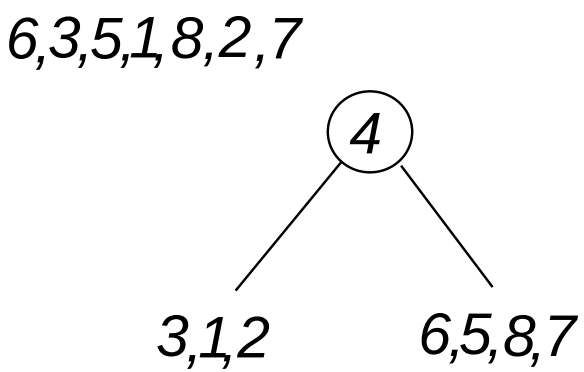
Quicksort

Sortieren von Daten

4,6,3,5,1,8,2,7

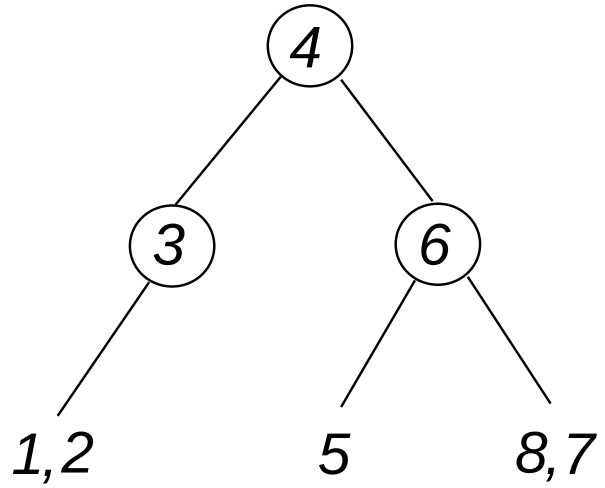
Quicksort

Sortieren von Daten



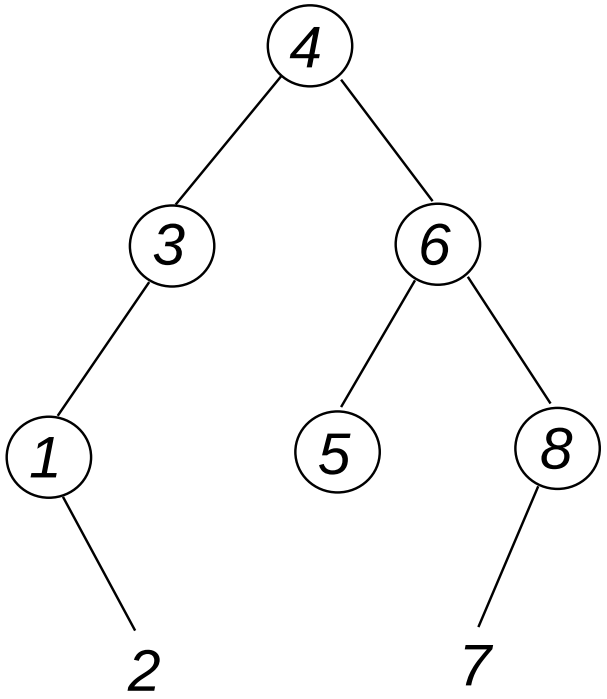
Quicksort

Sortieren von Daten



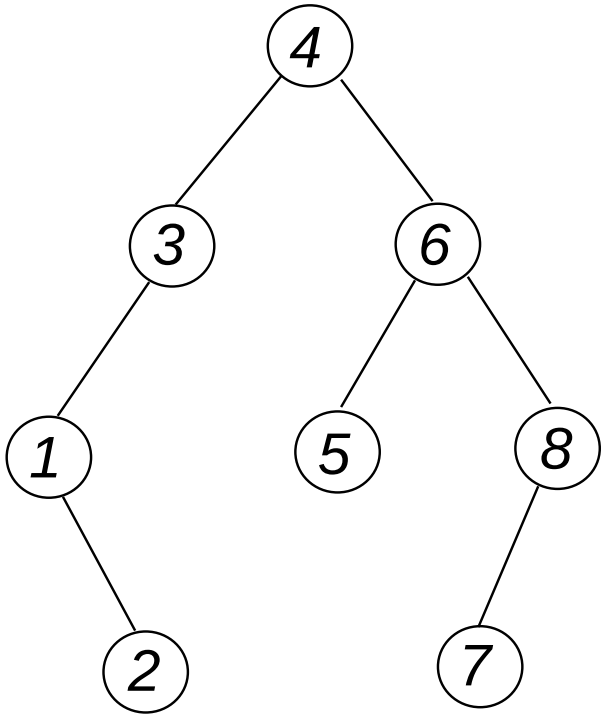
Quicksort

Sortieren von Daten



Quicksort

Sortieren von Daten



Binäre Suchbäume

Speichern von Daten

4,6,3,5,1,8,2,7

Binäre Suchbäume

Speichern von Daten

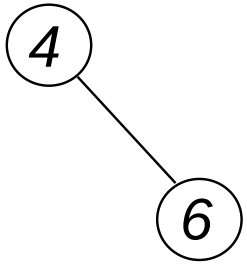
6,3,5,1,8,2,7

4

Binäre Suchbäume

Speichern von Daten

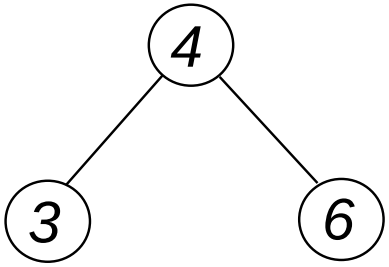
3,5,1,8,2,7



Binäre Suchbäume

Speichern von Daten

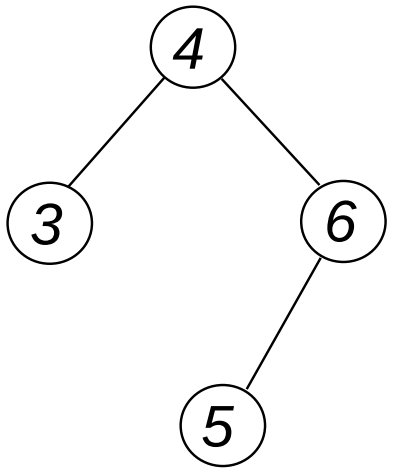
5,1,8,2,7



Binäre Suchbäume

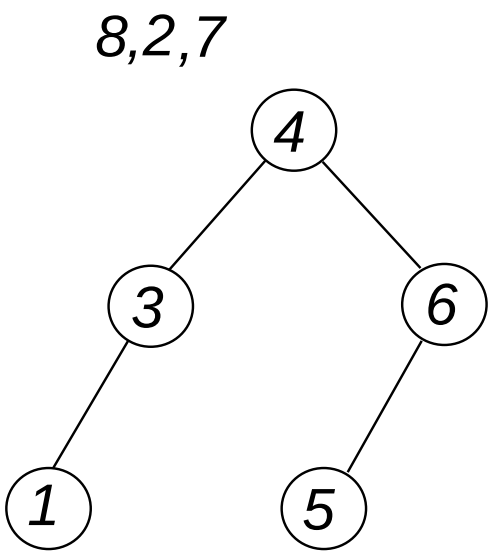
Speichern von Daten

1,8,2,7



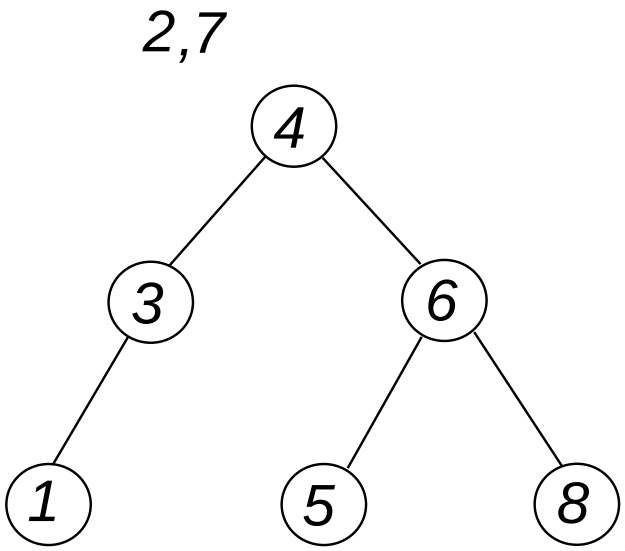
Binäre Suchbäume

Speichern von Daten



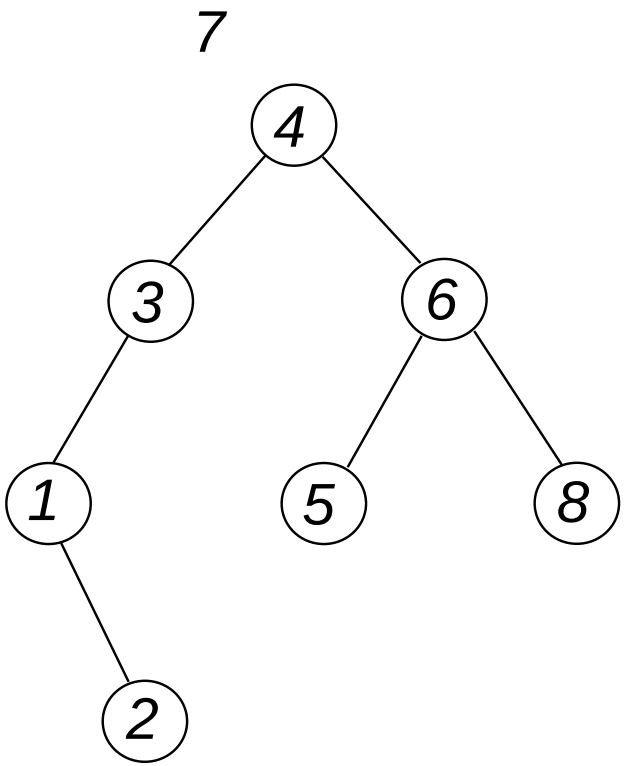
Binäre Suchbäume

Speichern von Daten



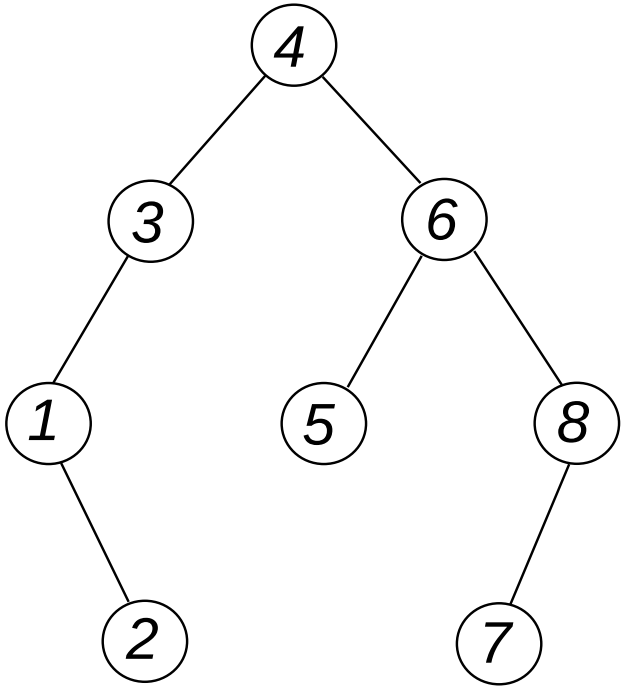
Binäre Suchbäume

Speichern von Daten



Binäre Suchbäume

Speichern von Daten



Quicksort

Wahrscheinlichkeisttheoretisches Modell

Jede Permutation der Daten $\{1, 2, \ldots, n\}$ ist gleichwahrscheinlich.

Zahl der Vergleichsopertationen

L_n ... Anzahl der Vergleichsopertatonen, um eine zufällige Permutation von $\{1, 2, \ldots, n\}$ mit Quicksort zu sortieren.

$$\boxed{L_n = \; L_{Z_n - 1} + \overline{L}_{n - Z_n} + n - 1}, \qquad n \geq 2,$$

$L_0 = L_1 = 0$, $L_2 = 1$, Z_n ist gleichverteilt auf $\{1, 2, \ldots, n\}$, $\overline{L}_j$ ist eine unabhängige Kopie von L_j und Z_n, L_j, $\overline{L}_j$ $(1 \leq j \leq n)$ sind unabhängig.

Quicksort

Erwartete Anzahl von Vergleichen

$$\mathbf{E}\, L_n = n - 1 + \frac{1}{n} \sum_{j=1}^{n} \left(\mathbf{E}\, L_{j-1} + \mathbf{E}\, L_{n-j} \right)$$

$\Longrightarrow$

$$\begin{aligned} \mathbf{E}\, L_n &= 2(n+1) \sum_{h=1}^{n+1} \frac{1}{h} - 4(n+1) + 2 \\ &= 2n \log n + n(2\gamma - 4) + 2 \log n + 2\gamma + 1 + \mathcal{O}\left((\log n)/n\right), \end{aligned}$$

$\gamma = 0.57721...$ Eulerkonstante.

Quicksort

Theorem. [Régnier, Rösler]

Die normalisierte Zahl der Vergleiche

$$\boxed{Y_n = \frac{L_n - \mathbf{E}\ L_n}{n}}$$

konvergiert zu einer Zufallsvariablen Y:

$$\boxed{Y_n \to Y},$$

die die stochastische Fixpunktgleichung

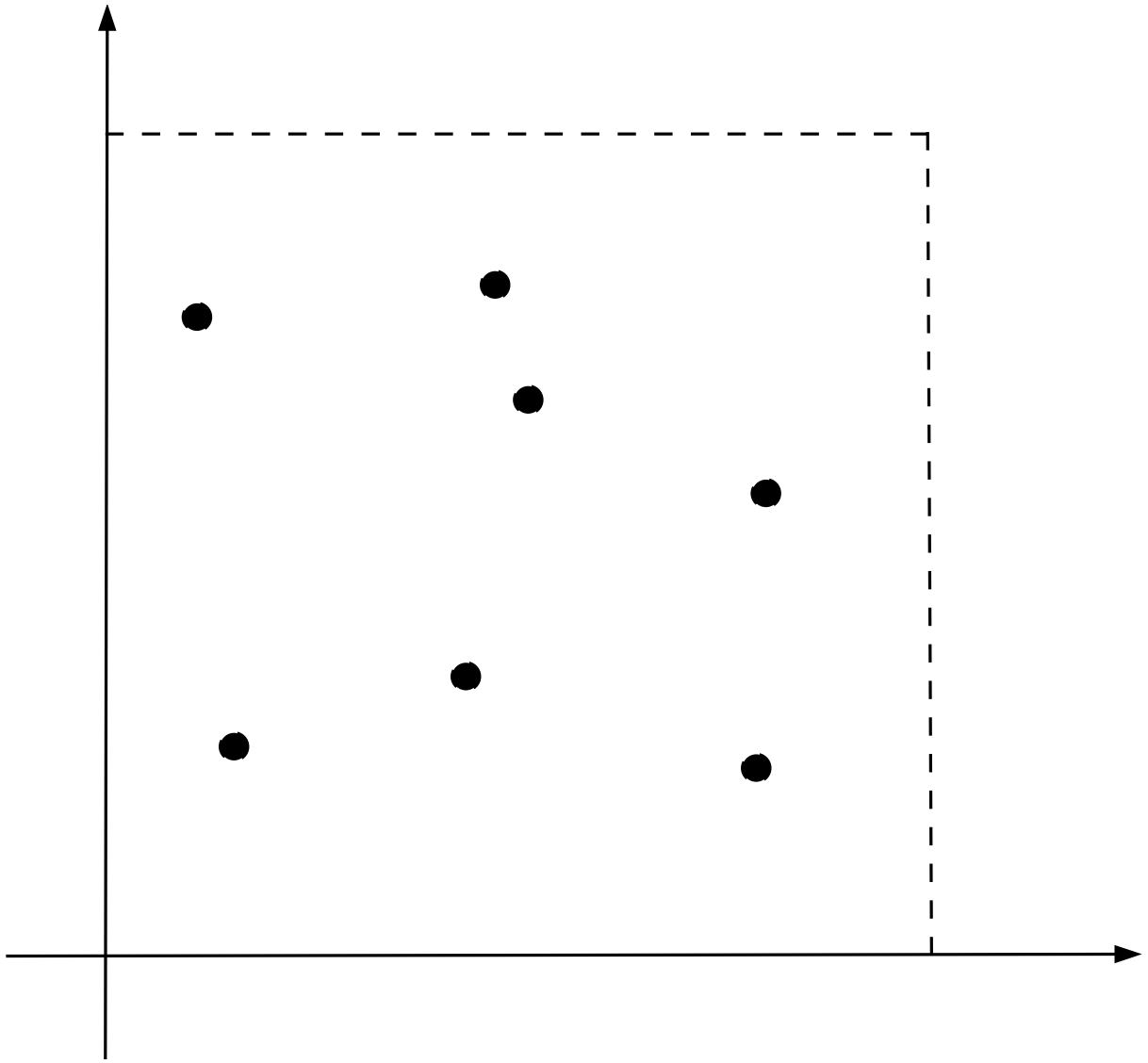
$$\boxed{Y = UY + (1-U)\overline{Y} + c(U)}$$

erfüllt, U ist gleichverteilt auf $[0,1]$, $\overline{Y}$ ist eine unabhängige Kopier von Y, $U, \overline{Y}, Y$ sind unabhängig und

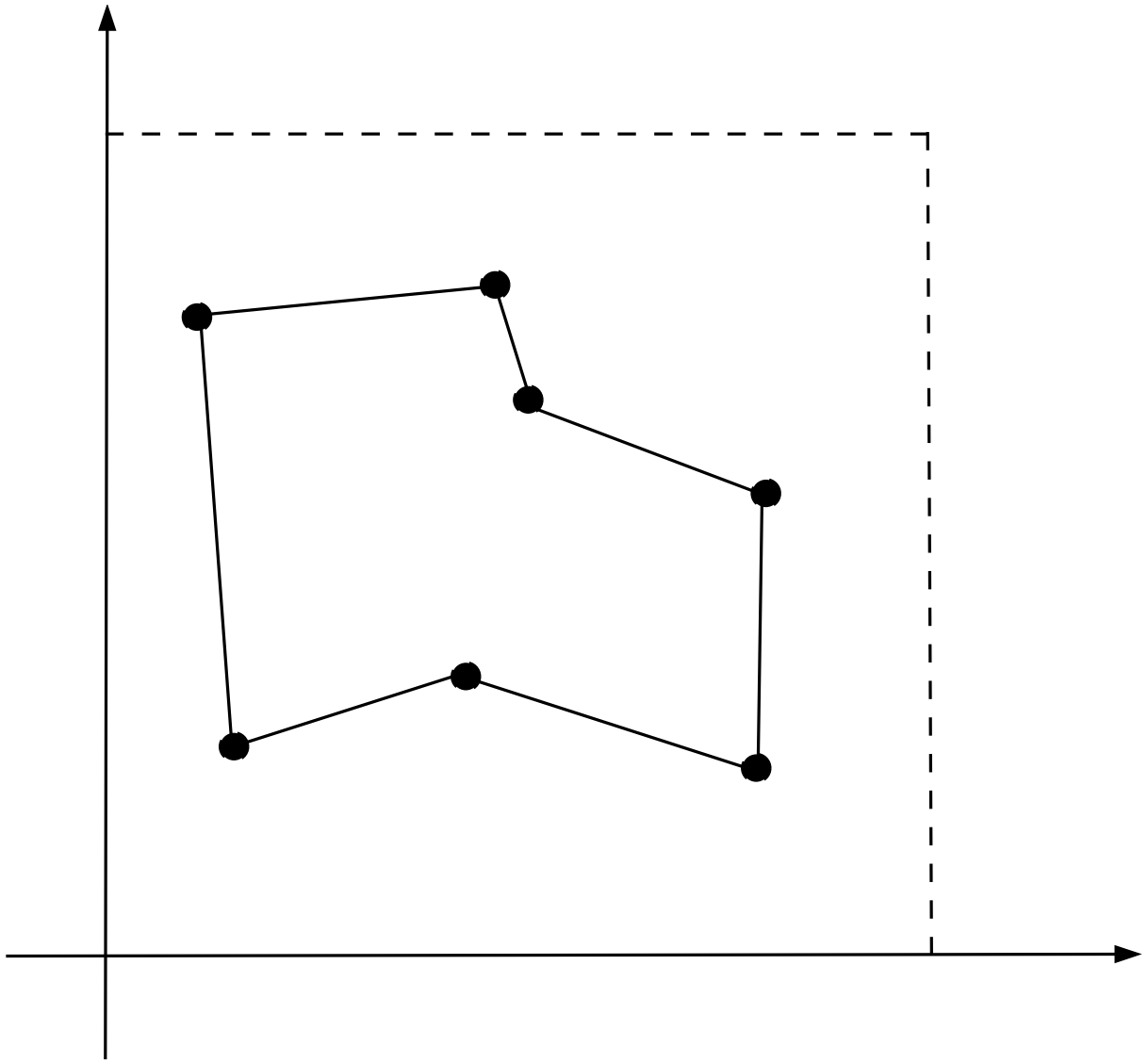
$$c(x) = 2x \log x + 2(1-x)\log(1-x) + 1.$$

(Y ist nicht normalverteilt.)

Travelling Salesman Problem



Travelling Salesman Problem



Travelling Salesman Problem

$\mathbf{X} = (X_1, X_2, \ldots, X_n)$ ... n-Tupel von zufällig gewählten Punkten im Einheitsquadrat $[0,1]^2$ (gleichverteilt und unabhängig).

Länge der minimalen Travelling Salesman-Tour:

$$\boxed{\mathrm{TSP}(\mathbf{X}) = \min_{\pi \in S_n} \sum_{j=1}^{n} \left| X_{\pi(j)} - X_{\pi(j+1)} \right|}$$

Theorem (Beardwood, Halton and Hammersley 1959)

$$\boxed{\frac{\mathrm{TSP}(\mathbf{X})}{\sqrt{n}} \to \beta_2} \quad \text{in prob.}$$

für ein $\beta_2 > 0$.

Bemerkung: Bis jetzt ist der exakte Wert von β_2 unbekannt.

Travelling Salesman Problem

Notation: M(Y) ... Median einer Zufallsvariablen Y

Theorem (Rhee and Talagrand)

$$\boxed{\mathbf{Pr}\left\{|\mathrm{TSP}(\mathbf{X}) - \mathrm{M}(\mathrm{TSP}(\mathbf{X}))| \geq t\right\} < 4e^{-t^2/c}.}$$

mit einer Konstanten $c > 0$.

Folgerung. Alle zentralen Momente von TSP($\mathbf{X}$) sind beschränkt.