

WS 2005/06

Diskrete Strukturen

Ernst W. Mayr

Fakultät für Informatik
TU München

<http://www14.in.tum.de/lehre/2005WS/ds/>

17. Januar 2006

Anwendung: Sei eine homogene Rekursion gegeben, z. B.

$$F_{n+2} = F_{n+1} + F_n \quad F_0 = 0, F_1 = 1$$

- ① Drücke die Rekursion in einer einzigen Formel aus, inklusive der Anfangsbedingungen. Wie immer ist $F_n = 0$ für $n < 0$. $F_n = F_{n-1} + F_{n-2}$ gilt auch für $n = 0$, aber für $n = 1$ ist $F_1 = 1$, die rechte Seite jedoch 0. Also ist die vollständige Rekursion

$$F_n = F_{n-1} + F_{n-2} + \delta_{n,1},$$

mit

$$\delta_{n,m} = \begin{cases} 1 & n = m \\ 0 & \text{sonst} \end{cases}$$

- 2 Interpretiere die Gleichung aus 1. mit Hilfe von erzeugenden Funktionen. Wir wissen schon, dass Indexerniedrigung einer Multiplikation mit einer Potenz von z entspricht. Also erhalten wir:

$$\begin{aligned} F(z) &= \sum F_n z^n \\ &= \sum F_{n-1} z^n + \sum F_{n-2} z^n + \sum \delta_{n,1} z^n \\ &= z \cdot F(z) + z^2 \cdot F(z) + z \end{aligned}$$

- 3 Löse die Gleichung in $F(z)$. Das ist leicht:

$$F(z) = \frac{z}{1 - z - z^2}$$

- ④ Drücke die rechte Seite als formale Reihe aus und ermittle daraus die Koeffizienten. Dies ist der schwierigste Schritt. Zunächst schreiben wir $1 - z - z^2$ in der Form $1 - z - z^2 = (1 - \alpha z)(1 - \beta z)$ und ermitteln dann durch Partialbruchzerlegung die Konstanten a und b , so dass gilt:

$$\frac{1}{(1 - \alpha z)(1 - \beta z)} = \frac{a}{1 - \alpha z} + \frac{b}{1 - \beta z} .$$

Es ergibt sich z.B.

$$\alpha = \frac{1 + \sqrt{5}}{2} \quad \beta = \frac{1 - \sqrt{5}}{2}$$

Es gilt:

$$\begin{aligned} F(z) &= z \left(\frac{a}{1 - \alpha z} + \frac{b}{1 - \beta z} \right) \\ &= z \left(a \sum_{n \geq 0} \alpha^n z^n + b \sum_{n \geq 0} \beta^n z^n \right) \\ &= \sum_{n \geq 1} (a\alpha^{n-1} + b\beta^{n-1}) z^n \end{aligned}$$

und somit

$$\begin{aligned} F_n &= a\alpha^{n-1} + b\beta^{n-1} \\ &= \frac{1}{\sqrt{5}} \left(\left(\frac{1 + \sqrt{5}}{2} \right)^n - \left(\frac{1 - \sqrt{5}}{2} \right)^n \right), \end{aligned}$$

nachdem man die Konstanten a und b etwa aus den Gleichungen für F_0 und F_1 bestimmt hat.

4.12 Das Master-Theorem

Bei der Analyse von **Divide-and-Conquer**-Verfahren stößt man oft auf Rekursionen, die sich nicht als lineare Rekursionen formulieren lassen. So führt der Mergesort-Algorithmus in der Standardvariante zu der Rekursionsgleichung

$$C_n = C_{\lfloor n/2 \rfloor} + C_{\lceil n/2 \rceil} + n \quad \text{für alle } n > 1 \quad \text{und } C_1 = 0.$$

Löst man allgemein ein Problem der Größe n dadurch, dass man es in a Teilprobleme der Größe höchstens n/b aufteilt, so erhält man für die Laufzeit $T(n)$ eine Rekursion der Form

$$T(n) = a \cdot T(n/b) + f(n),$$

wobei $f(n)$ die Laufzeit für die Aufteilung in Teilprobleme und für das Zusammenfügen der Lösungen der Teilprobleme ist.

Satz 224 (Master-Theorem)

Seien $a \in \mathbb{N}$, $b > 1$ und $C \geq 0$ Konstanten, und sei $f(n)$ eine nichtnegative Funktion. Weiter seien $c_1(n), \dots, c_a(n)$ Funktionen mit $|c_i(n)| \leq C$ für alle $1 \leq i \leq a$ und $n \in \mathbb{N}_0$. Ist dann $T(n)$ eine Funktion, die für $n = 1$ gleich 0 ist und die für $n \geq 1$ die Rekursionsgleichung

$$T(n) = T(n/b + c_1(n)) + \dots + T(n/b + c_a(n)) + f(n)$$

erfüllt, dann gilt

$$T(n) = \begin{cases} \Theta(n^{\log_b a}), & \text{falls } f(n) = O(n^{\log_b a - \epsilon}) \text{ für ein } \epsilon > 0, \\ \Theta(n^{\log_b a} \log n), & \text{falls } f(n) = \Theta(n^{\log_b a} \cdot \log^\delta n) \text{ f. } \delta > 0, \\ \Theta(f(n)), & \text{falls } f(n) = \Omega(n^{\log_b a + \epsilon}) \text{ für ein } \epsilon > 0. \end{cases}$$

Für den Beweis des Master-Theorems verweisen wir auf die Literatur, z.B. in:

- 1 Verma, Rakesh M.:

A general method and a master theorem for divide-and-conquer recurrences with applications.

J. Algorithms **16**, 1 (1994), pp. 67–79

- 2 Roura, Salvador:

An improved master theorem for divide-and-conquer recurrences.

Proceedings of the 24th International Colloquium on Automata, Languages and Programming, ICALP'97 (Bologna, Italy, July 7-11, 1997). LNCS **1256** (1997), pp. 449–459

Satz 225 (“Baby-Version” des MT)

Wenn die Funktion T für $x < 1$ gleich 0 ist und wenn für $x \geq 1$ die Rekursion

$$T(x) = aT(x/b) + x$$

gilt (also $T(1) = 1$), dann gilt für $n = b^t$ eine ganzzahlige Potenz von b :

$$T(n) = (1 + o(1)) \cdot \begin{cases} \frac{b}{b-a}n, & \text{falls } a < b, \\ n \log_b n, & \text{falls } a = b, \\ \frac{a}{a-b}n^{\log_b a}, & \text{falls } a > b. \end{cases}$$

Beweis:

Zuerst wenden wir die Rekursionsgleichung so oft an, bis wir die Anfangsbedingung erreichen. Wir haben also

$$\begin{aligned}T(n) &= n + aT(n/b) \\&= n + a\frac{n}{b} + a^2T(n/b^2) \\&= n + a\frac{n}{b} + a^2\frac{n}{b^2} + a^3T(n/b^3) \\&= \dots \\&= n + a\frac{n}{b} + a^2\frac{n}{b^2} + \dots + a^tT(n/b^t),\end{aligned}$$

wobei $t = \log_b n$. Also

$$T(n) = n \left(1 + \frac{a}{b} + \dots + \frac{a^t}{b^t} \right)$$

Fallunterscheidung:

$a < b$: In diesem Fall konvergiert die Summe und wir erhalten:

$$T(n) \leq n \sum_{k \geq 0} \left(\frac{a}{b}\right)^k = \frac{b}{b-a} n .$$

$a = b$: In diesem Fall ist die Lösung

$$T(n) = n (\log_b n + 1) = (1 + o(1)) \cdot n \log_b n .$$

$a > b$: Wir erhalten:

$$\begin{aligned} T(n) &= n \left(\frac{a}{b}\right)^t \left(1 + \frac{b}{a} + \dots + \frac{b^t}{a^t}\right) \\ &\leq n \frac{a}{a-b} \left(\frac{a}{b}\right)^t \\ &= \frac{a}{a-b} a^{\log_b n} \\ &= \frac{a}{a-b} n^{\log_b a}, \end{aligned}$$

da $t = \log_b n$.



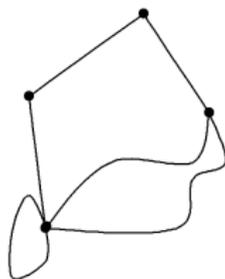
Kapitel IV Graphen und Algorithmen

1. Grundlagen

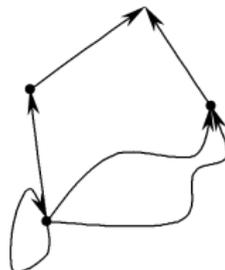
Definition 226

Ein Graph $G = (V, E)$ besteht aus einer Menge V von Knoten (aka Ecken, engl. **vertex**, **vertices**) und einer (Mehrfach-)Menge $E \subseteq V \times V$ von Paaren $(u, v) \in V \times V$, genannt Kanten (engl. **edges**).

Ein Graph heißt **ungerichteter** Graph, falls für alle $(u, v) \in E$ auch $(v, u) \in E$ ist. Man schreibt dann E auch als Menge von ungeordneten Paaren $\{u, v\}$ von Kanten.



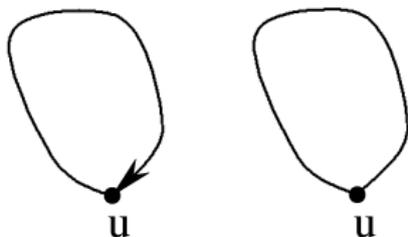
Ein Graph heißt ein **gerichteter** Graph, falls E (wie in obiger Definition) eine Menge von geordneten Paaren (u, v) ist.



1.1 Schlingen

Definition 227

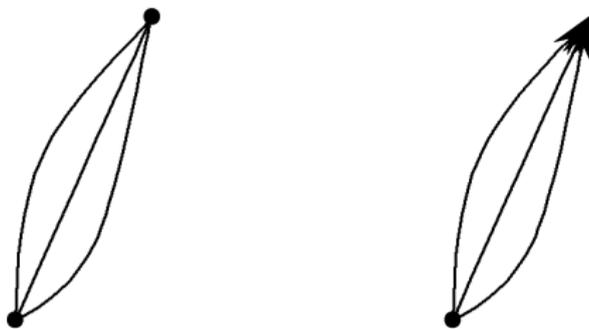
Eine **Schlinge** ist eine Kante der Form (u, u) bzw. $\{u, u\}$.



1.2 Mehrfachkanten

Definition 228

Ist E eine Multimenge (d. h. Kanten treten mit Vielfachheit auf), sind die Kanten mit Vielfachheit 2 oder größer **Mehrfachkanten**.



Ein Graph, der Mehrfachkanten enthält, heißt auch **Multigraph**.

1.3 Einfache Graphen

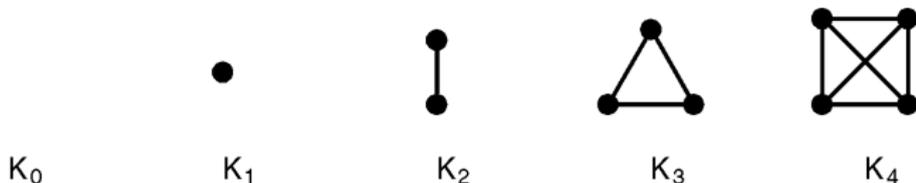
Definition 229

Ein Graph heißt **einfach**, falls er keine Schlingen oder Mehrfachkanten enthält.

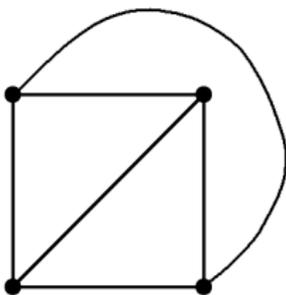
Definition 230

Ein Graph $G = (V, E)$ ($=: K_n$) mit $|V| = n$ Knoten heißt **vollständig** (**der vollständige Graph mit n Knoten**), falls $E = \{\{u, v\}; u, v \in V, u \neq v\}$ bzw. $E = \{(u, v); u, v \in V, u \neq v\}$.

Beispiel 231



Der K_4 lässt sich auch kreuzungsfrei zeichnen:



Für die Anzahl der Kanten in einem vollständigen Graphen (und damit für die **maximale** Anzahl von Kanten in einem einfachen Graphen) gilt:

$$|E| = \binom{n}{2} = \frac{n \cdot (n - 1)}{2}$$

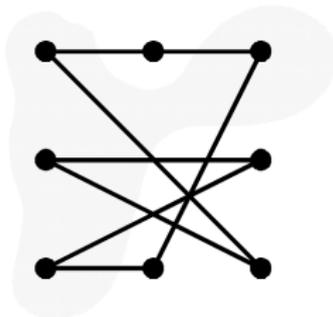
1.4 Bipartiter Graph

Definition 232

Ein Graph heißt **bipartit**, falls sich V in $V_1 \uplus V_2$ mit $V_1 \neq \emptyset \neq V_2$ so partitionieren lässt, dass gilt:

$$(\forall e \in E) [e \in (V_1 \times V_2) \cup (V_2 \times V_1)]$$

Beispiel 233 (C_8 , Kreis mit 8 Knoten)



Bemerkung:

Schreibweise für bipartite Graphen:

$$G = (V_1, V_2, E)$$

1.5 Vollständiger bipartiter Graph

Definition 234

Ein bipartiter Graph $G = (V_1, V_2, E)$ heißt **vollständig**, falls $E = V_1 \times V_2 \cup V_2 \times V_1$.

(Notation: $K_{m,n}$ mit $m = |V_1|, n = |V_2|$)

Beispiel 235



$K_{1,1}$



$K_{1,2}$



$K_{3,3}$

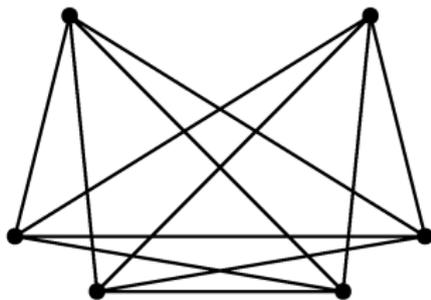
1.6 k -partiter Graph

Definition 236

Ein Graph heißt k -partit ($k \in \mathbb{N}, k \geq 2$), falls es eine Partition $V = V_1 \uplus V_2 \uplus \dots \uplus V_k$ mit $V_i \neq \emptyset, i = 1, \dots, k$ gibt, so dass

$$(\forall e \in E) [e \in V_i \times V_j; 1 \leq i, j \leq k, i \neq j]$$

Beispiel 237 (Vollständiger tripartiter Graph $K_{2,2,2}$)



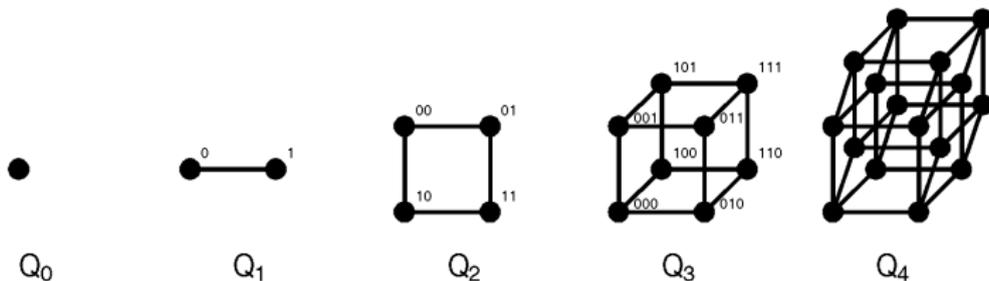
1.7 (Binärer) Hyperwürfel

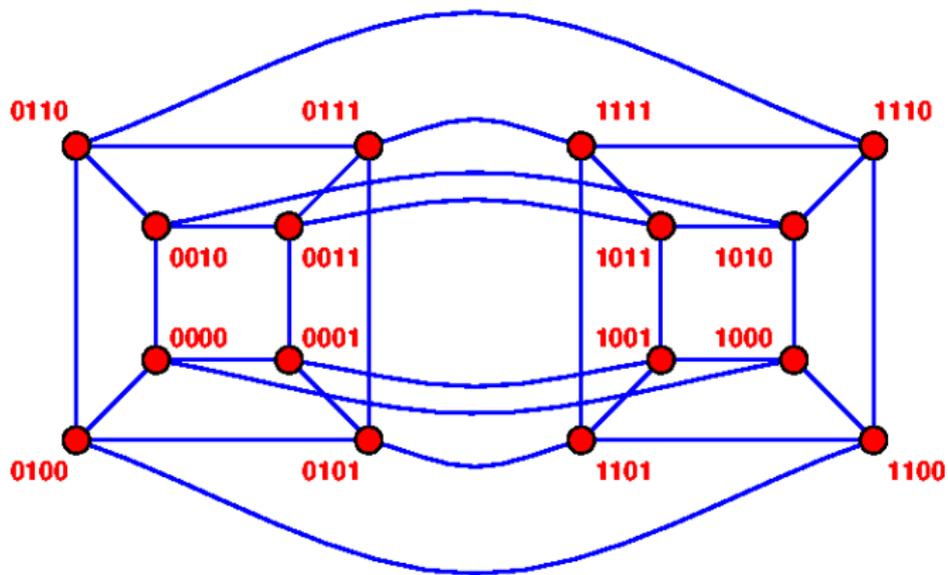
Definition 238

Ein Graph $G = (V, E)$ heißt n -dimensionaler binärer Hyperwürfel (aka Q_n), falls $V = V_n = \{0, 1\}^n$ mit

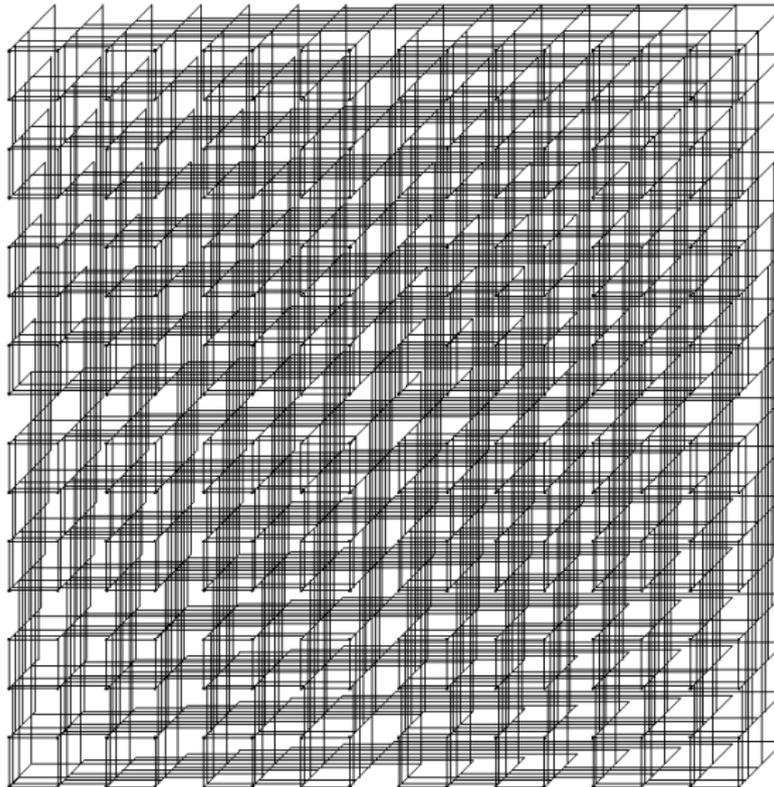
$$E = \left\{ \{v, w\} \in V_n^2; \text{Hamming-Abstand}(v, w) = 1 \right\}.$$

Beispiel 239





Q_4 : 4-dimensionaler Hyperwürfel



Q_8 : 8-dimensionaler Hyperwürfel

Für die Anzahl der Knoten in Q_n gilt:

$$|V| = 2^n$$

Für die Anzahl der Kanten in Q_n gilt:

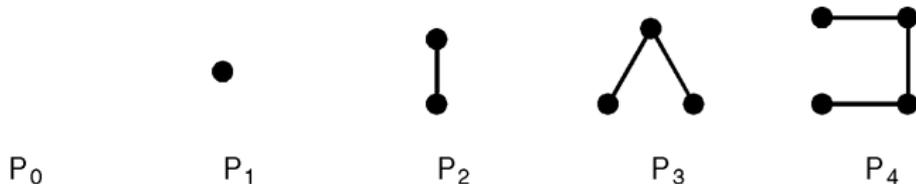
$$|E| = n \cdot \frac{2^n}{2} = n \cdot 2^{n-1}$$

1.8 Pfade

Definition 240

- 1 Ein Pfad der Länge n ist eine Folge (v_1, v_2, \dots, v_n) von Knoten eines Graphen $G = (V, E)$, so dass $(v_i, v_{i+1}) \in E$ für alle $i = 1, \dots, n - 1$.
- 2 Der Graph P_n ist der Graph (V, E) mit $V = \{v_1, \dots, v_n\}$ und $E = \{(v_i, v_{i+1}); i = 1, \dots, n - 1\}$.

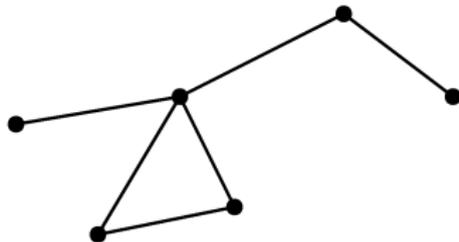
Beispiel 241



Definition 242

Ein Pfad heißt **einfach**, falls alle Knoten paarweise verschieden sind.

Beispiel 243 (Pfad, aber *nicht einfacher* Pfad der Länge 7)



1.9 Kreise

Definition 244

Ein Graph $G = (V, E)$ heißt (einfacher) Kreis der Länge n (i. Z. $C_n, n \geq 3$), falls $V = \{v_0, \dots, v_{n-1}\}$ und $E = \{\{v_i, v_{(i+1) \bmod n}\}; i = 0, \dots, n - 1\}$.

1.10 Gitter

Definition 245

Ein Graph $G = (V, E)$ heißt ein m - n -Gitter (zweidimensionales Gitter mit den Seitenlängen m und n , i. Z. $M_{m,n}$), falls $V = \{1, \dots, m\} \times \{1, \dots, n\}$ und

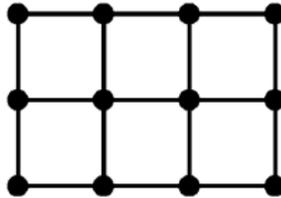
$$\underbrace{\{(i, j), (k, l)\}} \in E \iff |i - k| + |j - l| = 1$$

Kante zwischen
Knoten (i, j)
und Knoten (k, l)

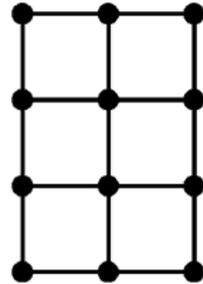
Beispiel 246



$M_{1,2}$



$M_{3,4}$



$M_{4,3}$

1.11 Torus

Definition 247

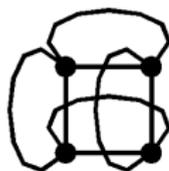
Ein Graph $G = (V, E)$ heißt **zweidimensionaler Torus** (pl. Tori) mit den Seitenlängen m und n , falls $V = \{1, \dots, m\} \times \{1, \dots, n\}$ und

$$\{(i, j), (k, l)\} \in E \iff |i - k| \bmod m + |j - l| \bmod n = 1$$

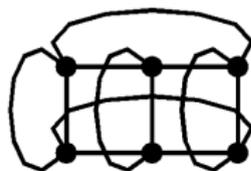
Beispiel 248



$T_{1,2}$



$T_{2,2}$



$T_{2,3}$

1.12 Petersen-Graph

Definition 249

Der folgende Graph heißt **Petersen-Graph**:

