I. Asymptotische Notation

(i) O-Notation „obere Schranke"

$f \in O(g(n)) \Leftrightarrow$ Es gibt ~~[illegible]~~ $c > 0$, $n_0 \in \mathbb{N}$, so dass
$0 \leq f(n) \leq c \cdot g(n) \quad \forall\, n \geq n_0$ gilt.

Beispiel:

(1.) $f(n) = 430n^3 + 175n^2 + 23$
$g(n) = n^3$

Dann:

$$f(n) = 430n^3 + 175n^2 + 23 \underset{n \geq 1}{\leq} 430n^3 + 175n^3 + 23$$

$$\underset{n \geq 1}{\leq} 430n^3 + 175n^3 + 23n^3 = 628n^3 \underset{\text{für } c = 628}{=} c \cdot g(n)$$

Damit:

Für $c = 628$ und $n \geq n_0 = 1$ gilt
$f(n) \leq c \cdot g(n)$

Zudem: $f(n) = 430n^3 + 175n^2 + 23 \geq 0$

Also:

$0 \leq f(n) \leq c \cdot g(n) \quad \forall\, n \geq n_0 = 1$.

(ii) Ω- Notation „untere Schranke" II

$\uparrow$ „omega"

$f \in \Omega(g(n)) \Leftrightarrow$ Es gibt $c > 0$, $n_0 \in \mathbb{N}$, so dass $0 \leq c \cdot g(n) \leq f(n)$ $\forall n \geq n_0$ gilt.

Beispiel:

(2) $f(n) = 430n^3 + 175n^2 + 23$
$g(n) = n^3$

Dann:

$$g(n) = 1 \cdot n^3 \underset{n \geq 1}{\leq} 430n^3 \underset{n \geq 1}{\leq} 430n^3 + 175n^2 + 23 = f(n)$$

Damit:

Für $c = 1$, $n_0 = 1$ gilt:

$$c \cdot g(n) \leq f(n) \quad \forall n \geq n_0$$

Zudem: $g(n) \geq 0$

Also:

$$0 \leq c \cdot g(n) \leq f(n) \quad \text{für } c = 1, n \geq n_0 = 1$$

(iii) Θ - Notation

„Teta"

$f \in \Theta(g(n)) \Leftrightarrow$ Es gibt $c_1, c_2 > 0$, $n_0 \in \mathbb{N}$, so dass

$0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \quad \forall n \geq n_0$ gilt.

Beispiel:

(3.) $f(n) = 430n^3 + 175n^2 + 23$

$g(n) = n^3$

Wie in Beispiel 1 und 2 gezeigt gilt:

Für $c_1 = 1$ und $c_2 = 628$, sowie $n_0 = 1$

$0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$

(4.) $f(n) = 13n^2 + 8$

$g(n) = 18n^2 + 3n$

(i) $g(n) \geq 0 \quad \forall n \geq 1$

(ii) $f(n) = 13n^2 + 8 \underset{n \geq 1}{\leq} 13n^2 + 8n^2 = 21n^2$

$\underset{n \geq 1}{\leq} 2 \cdot (18n^2 + 3n) = 2 \cdot g(n) \quad \Rightarrow c_2 = 2$

(iii) $f(n) = 13n^2 + 8 \overset{n \geq 1}{\geq} n^2 = \frac{1}{21}(21n^2) = \frac{1}{21}(18n^2 + 3n^2)$

$\geq \frac{1}{21}(18n^2 + 3n) = \frac{1}{21} g(n)$

$\Rightarrow c_1 = \frac{1}{21}$

N

Insgesamt haben wir:

$$0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \quad \forall n \geq n_0$$

für $c_1 = \frac{1}{21}$, $c_2 = 2$, $n_0 = 1$

Also: $f \in \Theta(g(n))$

$f(n) \leq c_1 \cdot g(n)$

$f(n) =$

$g(n) = 1$

$10n + 750 = 20n$

$\Leftrightarrow \cancel{750 = 10n}$

$\Leftrightarrow n = 75$

$n = 100$:

2000

1000

II. „Kodierung" von Zahlen im Computer

- Zahl $n \in \mathbb{N}$ im Computer speichern
- Binärsystem $\rightarrow$ Zahl zur Basis 2

Bsp.:

$n = 50 \qquad n = 5 \cdot 10 + 0 \cdot 1$

$= 0 \cdot 100 + 5 \cdot 10 + 0 \cdot 1$

$= 0 \cdot 10^2 + 5 \cdot 10^1 + 0 \cdot 10^0$

Ziffern einer Zahl geben an, wie oft bestimmte Zehner-Potenz verwendet wird, z.B.:

$$\underset{10^3\,10^2\,10^1\,10^0}{2537} = 2 \cdot 10^3 + 5 \cdot 10^2 + 3 \cdot 10^1 + 7 \cdot 10^0$$

$\hookrightarrow$ Man nennt 10 die „Basis"
und wir verwenden nur die Ziffern 0, 1, …, 9

D.h.:
Im Binärsystem ist die Basis 2
und wir verwenden nur die Ziffern 0 und 1.

Und wie rechnen wir das um??

$$(50)_{10} = (?)_2$$

„zur Basis 10"

$2^0 = 1 \leftarrow 0$

$2^1 = 2 \leftarrow 1$

$2^2 = 4 \leftarrow 0$

$2^3 = 8 \leftarrow 0$

$2^4 = 16 \leftarrow 1$

$2^5 = 32 \leftarrow 1$

$2^6 = 64$

d.h. $(50)_{10} = (\underset{2^5\,2^4\,2^3\,2^2\,2^1\,2^0}{110010})_2$

pro 0 oder 1 ein „Bit"

Hier brauchen wir also 6 Bits.

n=32: $(32)_{10} = (\overset{2^5\,2^4\,2^3\,2^2\,2^1\,2^0}{100000})_2$

6 Bits

$\log_2 32 = 5$, d.h. man braucht $\log_2 32 + 1$ Bits.

Allgemein: Für eine Zahl n braucht man höchstens

$\log n + 2$ Bits.

Begründung:

Für jedes n gibt es ein k mit

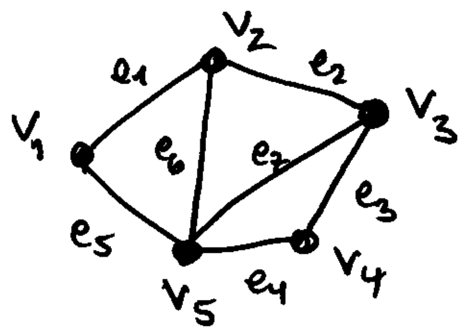
$$2^{k-1} < n \leq 2^k \iff \underbrace{k-1 < \log_2 n}_{k < \log_2 n + 1} \leq k$$

Für n braucht man also höchstens so viele Bits wie für 2^k.

Für 2^k braucht man $k+1$ Bits, d.h. für n höchstens

$$k+1 \leq (\log_2 n + 1) + 1 = \log_2 n + 2 \text{ Bits.}$$

III Datenstrukturen für Graphen



(1) Adjazenzmatrix

$$\begin{array}{c|ccccc} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \hline v_1 & 0 & 1 & 0 & 0 & 1 \\ v_2 & 1 & 0 & 1 & 0 & 1 \\ v_3 & 0 & 1 & 0 & 1 & 1 \\ v_4 & 0 & 0 & 1 & 0 & 1 \\ v_5 & 1 & 1 & 1 & 1 & 0 \end{array}$$

Größe: n^2

n Knoten

(2) Inzidenzmatrix

n Knoten, m Kanten

Größe: $n \cdot m$

$$\begin{array}{c|ccccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \\ \hline v_1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ v_2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ v_3 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ v_4 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ v_5 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}$$

(3) Adjazenzliste:

v_1: v_2 v_5
v_2: v_1 v_3 v_5
v_3: v_2 v_4 v_5
v_4: v_3 v_5
v_5: v_1 v_2 v_3 v_4

Muss je der Knotenindex kodiert werden.
Knotenindizes: $1, \dots, n$
Wir wissen: für das n braucht man $\log n + 2$ Bits
$\Rightarrow$ insgesamt höchstens $n \cdot (\log n + 2)$
also: $O(n \log n)$

(4) Kantenliste:

$e_1: v_1 v_2$

$e_2: v_2 v_3$

$\vdots$

Benötigt ebenfalls die Kodierung das Knotenindizes

$\Rightarrow$ höchstens $m \cdot (\lg n + 2)$

also: $O(m \lg n)$ Bits.