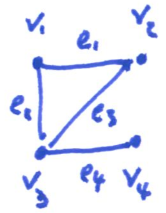
3.6 Datenstrukturen für Graphen

Wie beschreibt man einen Graphen?

(1) Inzidenzmatrix



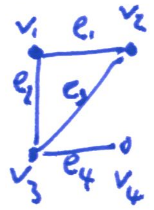
$$\begin{array}{c|cccc} & e_1 & e_2 & e_3 & e_4 \\ \hline v_1 & 1 & 1 & 0 & 0 \\ v_2 & 1 & 0 & 1 & 0 \\ v_3 & 0 & 1 & 1 & 1 \\ v_4 & 0 & 0 & 0 & 1 \end{array}$$

(„inzident": zusammentreffend)

Also: $A \in \{0,1\}^{n \times m}$ mit $a_{v,e} := \begin{cases} 1 & \text{für } v \in e \\ 0 & \text{sonst} \end{cases}$

Größe: nm für einen Graphen mit n Knoten, m Kanten. (Viele Nullen!)

(2) Adjazenzmatrix



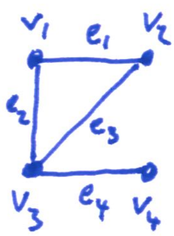
$$\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 1 & 1 & 0 \\ v_2 & 1 & 0 & 1 & 0 \\ v_3 & 1 & 1 & 0 & 1 \\ v_4 & 0 & 0 & 1 & 0 \end{array}$$

(„adjazent": verbunden)

Also: $A \in \{0,1\}^{n \times n}$ mit $a_{v,w} := \begin{cases} 1 & \text{für } \{v,w\} \in E \\ 0 & \text{sonst} \end{cases}$

Größe: n^2 für einen Graphen mit n Knoten.

(3) Kantenliste



$\{v_1,v_2\}, \{v_1,v_3\}, \{v_2,v_3\}, \{v_3,v_4\}$

Benötigt wird eine Kantennummerierung!

→ Jeder Index ist eine Binärzahl mit $(\lfloor \log_2 n \rfloor + 1)$ Bits, bzw. eine Dezimalzahl mit $(\lfloor \log_{10} n \rfloor + 1)$ Stellen.

Unterschied in Codierungslänge: Ein Faktor $\log_2 10 = 3{,}3219\ldots$,
denn $\log_2 n = \log_2 10 \cdot \log_{10} n$

Einschub: Wie viele Ziffern benötigt man für eine Zahl in Binär- oder Dezimaldarstellung?

Beispiele:

Binär	Dezimal	Stellen (binär)	(dezimal)
1	1	1	1
10	2	2	1
101	5	3	1
1111111111	1.023	10	2
1 0.000.000.000.000.000	65.536	17	5

Eine Zahl n mit b Binärstellen liegt zwischen 2^{b-1} und 2^b,
mit d Dezimalstellen liegt zwischen 10^{d-1} und 10^d:

$$2^{b-1} \leq n < 2^b$$

bzw. $$10^{d-1} \leq n < 10^d$$

Also gilt $$b-1 \leq \log_2 n < b$$

bzw. $$d-1 \leq \log_{10} n < d,$$

d.h. $$b = \lfloor \log_2 n \rfloor + 1$$

bzw. $$d = \lfloor \log_{10} n \rfloor + 1$$

Außerdem gilt mit $$2^b = \left(2^{\log_2 10}\right)^{\frac{b}{\log_2 10}} = 10^{\frac{b}{\log_2 10}},$$

also $d \approx \frac{1}{\log_2 10} b$, d.h. zwischen der binären und dezimalen Stellenzahl gibt es eine Umrechnungskonstante von $\log_2 10 \approx 3{,}3219\ldots$

(Das gilt nur ungefähr, weil wir ja ganze Zahlen haben und runden!)

Für einen Graphen mit m Kanten und n Knoten ergibt sich in obiger (ausführlicher) Codierung ein Platzbedarf von

$$(6m-1) + 2m\left(\lfloor \log_{10} n \rfloor + 1\right).$$

Dabei kann man an ein paar Stellen sparen (z.B. „v" oder „{", „}" weglassen) aber auch mehr Platz investieren (z.B. in ASCII codieren bzw. binär statt dezimal codieren). So wäre auch

$$(2m-1) + 2m\left(\lfloor \log_2 n \rfloor + 1\right) \quad \text{denkbar.}$$

Was ist wirklich wichtig dabei?!

(i) Die Kantenliste ist sparsamer als die Inzidenzmatrix, denn wenn n nicht zu klein ist, dann ist

$$n \geq 2 + 2\left(\lfloor \log_2 n \rfloor + 1\right).$$

(was heißt „nicht zu klein"? $n \geq 16$ reicht!)

Also ist auch

$$mn > (2m-1) + 2m\left(\lfloor \log_2 n \rfloor + 1\right).$$

(ii) Unabhängig von der Codierung ist für die Größe des Speicherplatzes der zweite Ausdruck wichtig, denn

$$\begin{aligned} 2m\left(\log_2 n + 1\right) &\leq (2m-1) + 2m\left(\lfloor \log_2 n \rfloor + 1\right) \\ &\leq 4m\left(\log_2 n + 1\right) \quad \text{für } n \geq 2. \end{aligned}$$

(iii) Letztlich kommt es also gar nicht so sehr auf die Vorfaktoren an (die sind codierungsabhängig!), sondern auf den Ausdruck

$$m \log n .$$

(iv) In $m \log n$ steckt das Wesen der Kantenliste: Zähle für alle m Kanten die Nummern der beteiligten Knoten auf!