4.4 Der Algorithmus von Johnson

Definition 113

Sei $\mathrm{d} : A \to \mathbb{R}$ eine Distanzfunktion. Eine Abbildung

$$r : V \to \mathbb{R}$$

heißt Rekalibrierung, falls gilt:

$$(\forall (u,v) \in A)[r(u) + \mathrm{d}(u,v) \geq r(v)]$$

Beobachtung: Sei r eine Rekalibrierung (für d). Setze $\mathrm{d}'(u,v) := \mathrm{d}(u,v) + r(u) - r(v)$. Dann gilt:

$$\mathrm{d}'(u,v) \geq 0$$

Sei $u = v_0 \to \cdots \to v_k = v$ ein Pfad. Dann ist:

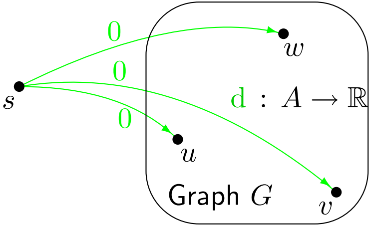
$$\mathrm{d}\text{-Länge} := \sum_{i=0}^{k-1} \mathrm{d}(v_i, v_{i+1})$$

Demnach ist:

$$\begin{aligned}
\mathrm{d}'\text{-Länge} &= \sum_{i=0}^{k-1} \mathrm{d}'(v_i, v_{i+1}) \\
&= \sum_{i=0}^{k-1} \left(\mathrm{d}(v_i, v_{i+1}) + r(v_i) - r(v_{i+1})\right) \\
&= \sum_{i=0}^{k-1} \mathrm{d}(v_i, v_{i+1}) + r(v_0) - r(v_k)
\end{aligned}$$

Also ist ein d-kürzester Pfad von u $(= v_0)$ nach v $(= v_k)$ auch ein d'-kürzester Pfad und umgekehrt. Nach einer Rekalibrierung kann man also auch die Algorithmen anwenden, die eine nichtnegative Distanzfunktion d voraussetzen (z.B. Dijkstra).

Berechnung einer Rekalibrierung:



Füge einen neuen Knoten s hinzu und verbinde s mit jedem anderen Knoten $v \in V$ durch eine Kante der Länge 0.

Berechne sssp von s nach allen anderen Knoten $v \in V$ (z.B. mit Bellman-Ford). Sei $r(v)$ die dadurch berechnete Entfernung von s zu $v \in V$. Dann ist r eine Rekalibrierung, denn es gilt:

$$r(u) + \mathrm{d}(u, v) \geq r(v)\,.$$

5. Zusammenfassung

	$d \geq 0$	d allgemein
sssp	D (Fibonacci): $\mathcal{O}(m + n \cdot \log n)$ D (Radix): $\mathcal{O}(m + n\sqrt{\log C})$	B-F: $\mathcal{O}(n \cdot m)$
apsp	D: $\mathcal{O}(n \cdot m + n^2 \min\{\log n, \sqrt{\log C}\})$ F: $\mathcal{O}(n^3)^{(*)}$	J: $\mathcal{O}(n \cdot m + n^2 \log n)$ F: $\mathcal{O}(n^3)$

Bemerkung$^{(*)}$: In der Praxis ist der Floyd-Algorithmus für kleine n besser als Dijkstra's Algorithmus.

6. Transitive Hülle

6.1 Min-Plus-Matrix-Produkt und Min-Plus-Transitive Hülle

Ring $\mathbb{Z}(+, \times)$ | Semiring $\mathbb{N}(+, \times)$

Gruppe | Halbgruppe | Halbgruppe | Halbgruppe

Wir betrachten den (kommutativen) Semiring über $\mathbb{R} \cup \{\infty\}$ mit den Operationen $\min$ und $+$. Für jede der beiden Operationen haben wir ein Monoid. Es gilt das Distributivgesetz
$a + \min\{b, c\} = \min\{a + b, a + c\}$.

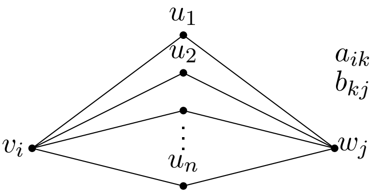
Normale Matrixmultiplikation:

$$A = (a_{ij})_{1 \le i,j \le n}, \quad B = (b_{ij})_{1 \le i,j \le n}, \quad I = (\delta_{ij})_{1 \le i,j \le n}$$

$$C = A \cdot B = (c_{ij})_{1 \le i,j \le n}, \quad c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$$

Entsprechend für Min-Plus:

$$c_{ij} = \min\{a_{ik} + b_{kj};\ 1 \le k \le n\}$$



$$a_{ik} = \mathrm{d}(v_i, u_k)$$
$$b_{kj} = \mathrm{d}(u_k, w_j)$$

Anwendung:
kürzeste Wege von v_i nach w_j in einem Graph ($A = B$); dabei ist

$$I_{\min,+} = \begin{pmatrix} 0 & & \infty \\ & \ddots & \\ \infty & & 0 \end{pmatrix}$$

Sei A Entfernungsmatrix, $A = (a_{ij})_{1 \le i,j \le n} = (\mathrm{d}(v_i, v_j))_{1 \le i,j \le n}$.
Setze $a_{ii} = 0$ für $i = 1, \ldots, n$.

Betrachte A^2 mit dem Min-Plus-Produkt, $A^2 =: (a_{ij}^{(2)})_{1 \le i,j \le n}$.

Dann ist $a_{ij}^{(2)}$ die Länge eines kürzesten Pfades von v_i nach v_j, der höchstens zwei Kanten enthält. Induktion ergibt: $a_{ij}^{(k)}$ ist die Länge eines kürzesten Pfades von v_i nach v_j mit höchstens k Kanten.
Falls die $\mathrm{d}(v_i, v_j)$ alle ≥ 0 sind, gibt es immer kürzeste Pfade, die höchstens $n - 1$ Kanten enthalten.

Damit ergibt sich folgende alternative Lösung des all-pairs-shortest-path-Problems:

Berechne A^{n-1} (Min-Plus)!

Es genügt auch, $A^{2^{\lceil \log(n-1) \rceil}}$ durch wiederholtes Quadrieren zu berechnen (nicht $A^2, A^3, A^4, \dots$).

Definition 114

$A^* := \min_{i \geq 0}\{A^i\}$ heißt Min-Plus-Transitive Hülle.

Bemerkung: $\min$ wird komponentenweise gebildet. Wenn $\mathrm{d} \geq 0$, dann $A^* = A^{n-1}$.

6.2 Boolesche Matrixmultiplikation und Transitive Hülle

Wir ersetzen nun im vorhergehenden Abschnitt die Distanzmatrix durch die (boolesche) Adjazenzmatrix und $(\min, +)$ durch $(\vee, \wedge)$, d.h.:

$$C = A \cdot B; \quad c_{ij} = \bigvee_{k=1}^{n} a_{ik} \wedge b_{kj}$$

Wenn wir zudem $a_{ii} = 1$ für $1 \leq i \leq n$ setzen, dann gilt für A^k (boolesches Produkt, $A^0 = I$)

$$a_{ij} = \begin{cases} 1 & \text{falls es im Graphen einen Pfad von } v_i \text{ nach } v_j, \\ & \text{bestehend aus } \leq k \text{ Kanten, gibt} \\ 0 & \text{falls es im Graphen keinen Pfad von } v_i \text{ nach } v_j, \\ & \text{bestehend aus } \leq k \text{ Kanten, gibt} \end{cases}$$

Transitive Hülle:

$$A^* := \bigvee_{i \geq 0} A^i \qquad (= A^{n-1})$$

ist damit die Adjazenzmatrix der transitiven Hülle des zugrunde liegenden Digraphen.

Satz 115

Sei $M(n)$ die Zeitkomplexität für das boolesche Produkt zweier $n \times n$-Matrizen, $T(n)$ die Zeitkomplexität für die transitive Hülle einer $n \times n$ booleschen Matrix.

Falls $\underbrace{T(3n) \le cT(n)}_{\substack{\text{sicher erfüllt, falls} \\ T \text{ polynomiell}}}$ *und* $\underbrace{M(2n) \ge 4M(n)}_{\substack{\text{sicher erfüllt, falls} \\ M(n) \ge n^2}}$*, dann gilt:*

$$T(n) = \Theta(M(n))\,.$$

Beweis:

(1) Matrixmultiplikation $\prec$ transitive Hülle:

Seien boolesche Matrizen A, B gegeben und ihr boolesches Produkt $C = A \cdot B$ gesucht.

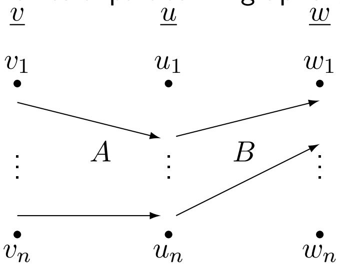
Setze:

$$L = \underbrace{\left.\begin{pmatrix} 0 & A & 0 \\ 0 & 0 & B \\ 0 & 0 & 0 \end{pmatrix}\right\}}_{3n} 3n$$

Beweis (Forts.):

L ist die Adjazenzmatrix eines tripartiten Digraphen, denn:

$$\begin{array}{c|ccc} & \underline{v} & \underline{u} & \underline{w} \\ \hline \underline{v} & 0 & A & 0 \\ \underline{u} & 0 & 0 & B \\ \underline{w} & 0 & 0 & 0 \end{array}$$



Daher kann L^* leicht bestimmt werden:

$$L^* = \begin{pmatrix} I & A & AB \\ 0 & I & B \\ 0 & 0 & I \end{pmatrix} \qquad (= I \vee L \vee L^2)$$

Also gilt: $M(n) \leq T(3n) = \mathcal{O}(T(n))$.

Beweis (Forts.):

(2) Transitive Hülle $\prec$ Matrixmultiplikation:
Gegeben: $n \times n$ boolesche Matrix L; gesucht: L^*; Annahme: n ist Zweierpotenz. Teile auf:

$$L = \begin{array}{cc} \begin{pmatrix} A & B \\ C & D \end{pmatrix} & \begin{array}{l} \}\, \frac{n}{2} \\ \}\, \frac{n}{2} \end{array} \\ \underbrace{\hphantom{A\ B}}_{\frac{n}{2}}\underbrace{\hphantom{C\ D}}_{\frac{n}{2}} & \end{array} ; \quad L^* = \begin{pmatrix} E & F \\ G & H \end{pmatrix}$$

Es gilt also:

$E =$	$(A \vee BD^*C)^*$	betrachte alle Pfade von der ersten Hälfte der Knoten zur ersten Hälfte
$F =$	EBD^*	analog
$G =$	D^*CE	analog
$H =$	$D^* \vee GF$	analog

Beweis (Forts.):

Um L^* zu berechnen, benötigen wir zwei Transitive-Hülle-Berechnungen und sechs Matrixprodukte für Matrizen der Dimension $\frac{n}{2} \times \frac{n}{2}$ (nämlich $M_1 = D^*C$, $M_2 = BM_1$, $M_3 = EB$, $M_4 = M_3D^*$, $M_5 = M_1E$, $M_6 = GF$), plus den Aufwand für $\vee$, der $\leq c'n^2$ ist. Wir zeigen nun durch Induktion ($n = 1\surd$), dass $T(n) \leq cM(n)$:

$$
\begin{aligned}
T(n) &\leq 2T(\tfrac{n}{2}) + 6M(\tfrac{n}{2}) + c'n^2 \\
&\leq 2cM(\tfrac{n}{2}) + 6M(\tfrac{n}{2}) + c'n^2 \quad && |\text{Vor.: } M(2n) \geq 4M(n) \\
& && |\qquad\quad \text{da } M(n) \geq n^2 \\
&\leq \tfrac{1}{4}(2c + 6 + 4c')M(n) \\
&\leq cM(n)
\end{aligned}
$$

falls $c \geq \frac{1}{4}(2c + 6 + 4c')$, also falls $c \geq 3 + 2c'$.

Also $T(n) = \mathcal{O}(M(n))$. $\square$