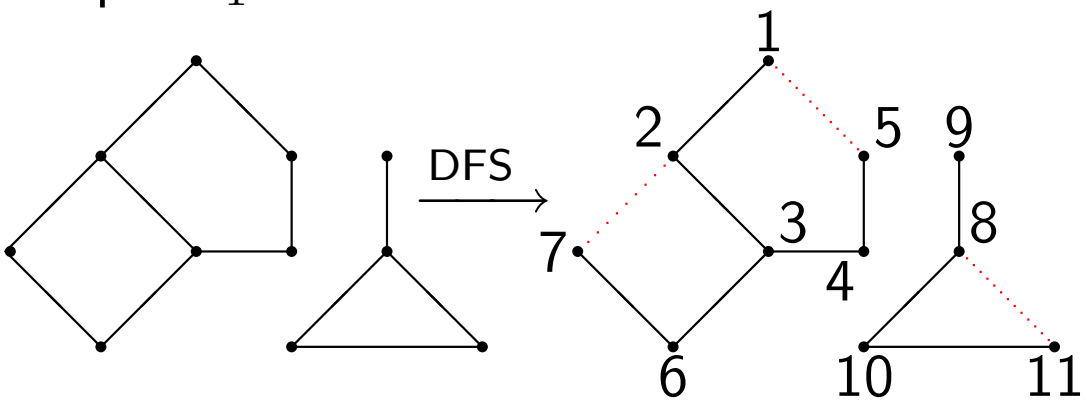
6.4 Transitive Hülle und DFS

Wir erinnern uns an den DFS-Algorithmus:

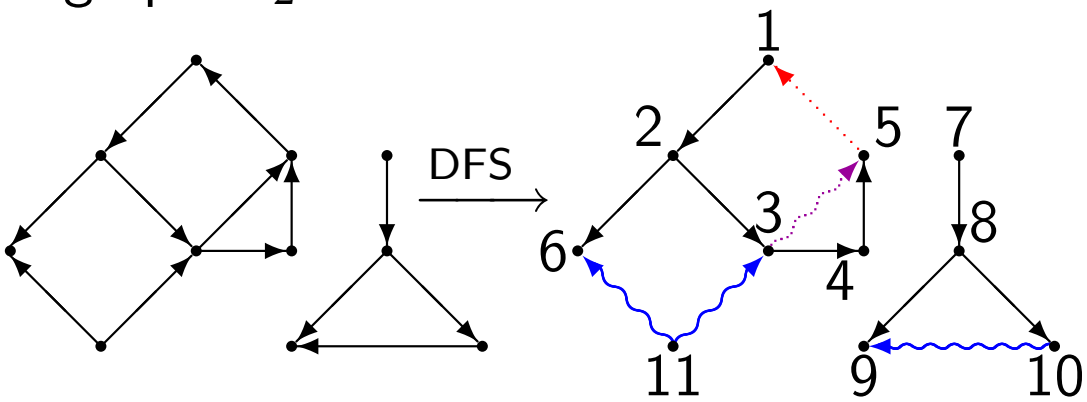
for all nodes v **do** unmark v **od**
$count := 0$
while $\exists$ unvisited v **do**
    $r :=$ pick (random) unvisited node
    push r onto stack
    while stack $\neq \emptyset$ **do**
        $v :=$ pop top element
        if v unvisited **then**
            mark v visited
            push all neighbours of v onto stack
            num[v]:= $++count$
        fi
    od
od

Beispiel 117

Graph G_1:



Digraph G_2:

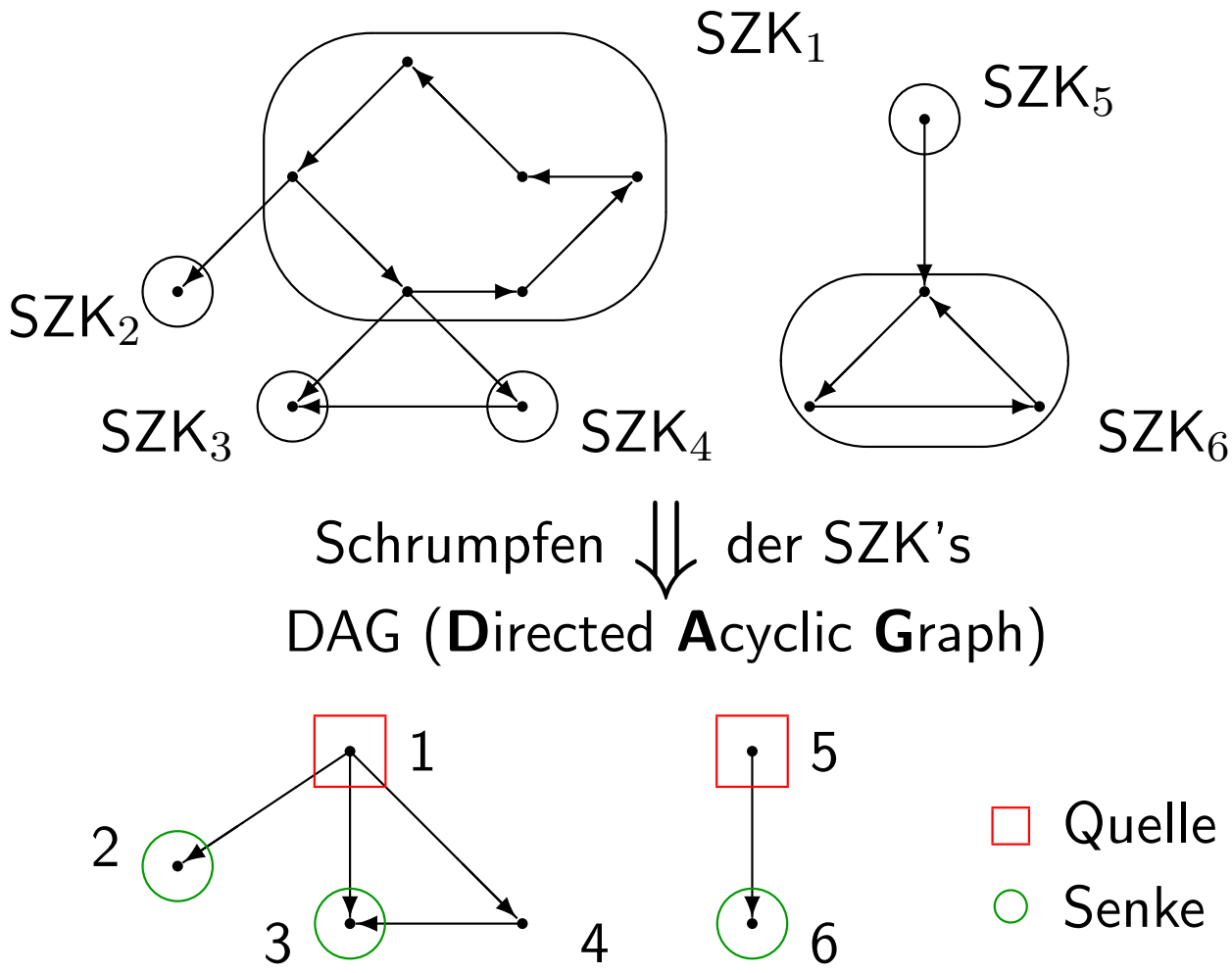


Bezeichnungen:

- Baumkante
- Rückwärtskante
- Querkante
- Vorwärtskante

Bem.: Alle Baumkanten zusammen: DFS-Wald (Spannwald).

Beispiel 118 (Starke Zusammenhangskomponenten (in Digraphen))

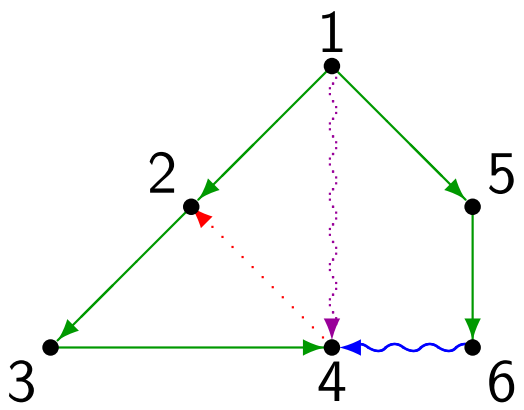


DFS in ungerichteten Graphen mit c Zusammenhangskomponenten, n Knoten, m Kanten: $n - c$ Baumkanten, $m - n + c$ Rückwärtskanten, Zeitaufwand $\mathcal{O}(n + m)$

$$\mathbf{A}^* \quad \begin{array}{c|c} & \begin{array}{cc} \mathsf{ZK}_1 & \mathsf{ZK}_2 \end{array} \\ \hline \mathsf{ZK}_1 & \left(\begin{array}{cc|c} \boxed{\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & & & 1 \\ 1 & & 1 & 1 \\ 1 & 1 & 1 & 1 \end{array}} & & 0 \\ & \boxed{} & \\ 0 & & \boxed{} \end{array}\right) \\ \mathsf{ZK}_2 & \end{array}$$

A^* aus DFS mit Aufwand $\Theta(n^2)$

DFS in gerichteten Graphen (Digraphen) mit n Knoten, m Kanten:
Baum-, Vorwärts-, Rückwärts- und Querkanten:



Bezeichnungen:

- Baumkanten
- Rückwärtskanten
- Querkanten
- Vorwärtskanten

Zeitaufwand: $\boxed{\mathcal{O}(n+m)}$

u ist von v aus auf einem gerichteten Pfad erreichbar gdw u Knoten in dem DFS-Baum (DFS-visit(v)) mit Wurzel v ist.

Also: Transitive Hülle in Zeit $\mathcal{O}(n \cdot (m+n))$. Sehr gut für dünne Graphen.

7. Ein besserer Algorithmus für das apsd-Problem in ungewichteten Digraphen

Sei $G = (V, E)$ ein Digraph mit der Knotenmenge $\{0, \ldots, n-1\}$, dessen Kanten alle die Länge 1 haben.

Sei $D = (d_{ij})_{0 \le i,j < n}$ die zugehörige Distanzmatrix, mit Einträgen $d_{ij} \in \{0, 1, \infty\}$.

Entsprechend sei D^* die Distanzmatrix, so dass

$$d^*_{ij} = \text{Länge eines kürzesten Wegs zwischen } i \text{ und } j$$

Setze weiterhin

$$D^{(l)} = (d^{(l)}_{ij})_{0 \le i,j < n} \text{ mit } d^{(l)}_{ij} = \begin{cases} d^*_{ij} & \text{falls } d^*_{ij} \le l \\ \infty & \text{sonst} \end{cases}$$

Algorithmus a1

co Sei $A = (a_{ij})_{0 \le i,j < n}$ mit $a_{ij} = \begin{cases} 1 \text{ falls } d_{ij} \le 1 \\ 0 \text{ sonst} \end{cases}$ **oc**

$B := I$ **co** boolesche Einheitsmatrix **oc**

for $l := 2$ **to** $r + 1$ **do**

  $B := A \cdot B$

  for $0 \le i, j < n$ **do**

    if $b_{ij} = 1$ **then** $d_{ij}^{(l)} := l$ **else** $d_{ij}^{(l)} := \infty$ **fi**

    if $d_{ij}^{(l-1)} \le l$ **then** $d_{ij}^{(l)} := d_{ij}^{(l-1)}$ **fi**

  od

od

Dieser Algorithmus berechnet $D^{(1)} = D$, $D^{(2)}$, $D^{(3)}, \ldots, D^{(r)}$.

Sei ω eine Zahl ≥ 2, so dass die Matrixmultiplikation in Zeit $\mathcal{O}(n^\omega)$ durchgeführt werden kann (Winograd/Coppersmith: $\omega \leq 2,376$).

Zeitaufwand für Algorithmus a1: $\boxed{\mathcal{O}(rn^\omega)}$

Algorithmus apsd =

Berechne, mit Algorithmus a1, $D^{(l)}$ für $l = 1, \ldots, r$; $l := r$

for $s := 1$ **to** $\left\lceil \log_{\frac{3}{2}} \frac{n}{r} \right\rceil$ **do**

  for $i := 0$ **to** $n - 1$ **do**

    finde in Zeile i von $D^{(l)}$ das d, $\left\lceil \frac{l}{2} \right\rceil \leq d \leq l$, das in dieser Zeile am wenigsten oft vorkommt

    $S_i :=$ Menge der zugehörigen Spaltenindizes

  od

  $l_1 := \left\lceil \frac{3}{2} l \right\rceil$

  for $0 \leq i, j < n$ **do**

    $m_{ij} :=$ **if** $S_i \neq \emptyset$ **then** $\min_{k \in S_i} \{d_{ik}^{(l)} + d_{kj}^{(l)}\}$ **else** ∞ **fi**

    if $d_{ij}^{(l)} \leq l$ **then** $d_{ij}^{(l_1)} := d_{ij}^{(l)}$

    elif $m_{ij} \leq l_1$ **then** $d_{ij}^{(l_1)} = m_{ij}$ **else** $d_{ij}^{(l_1)} := \infty$ **fi**

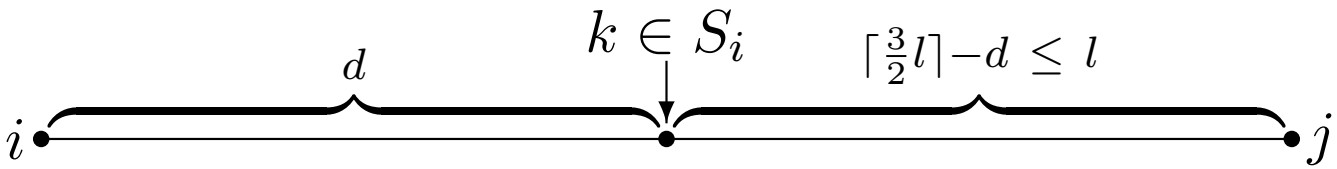
  od

  $l := l_1$

od

apsd berechnet
zunächst $D^{(1)}, \ldots, D^{(r)}$, dann $D^{(l)}$ für

$$l = r, \left\lceil \frac{3}{2} r \right\rceil, \left\lceil \frac{3}{2} \left\lceil \frac{3}{2} r \right\rceil \right\rceil, \ldots, \underbrace{n'}_{\geq n}$$



Da für d $\frac{l}{2}$ Werte zur Auswahl stehen, gilt:

$$|S_i| \leq \frac{2n}{l}$$

Damit ist die Laufzeit

$$\mathcal{O}\left(rn^\omega + \sum_{s=1}^{\left\lceil \log_{\frac{3}{2}} \frac{n}{r} \right\rceil} \frac{n^3}{\left(\frac{3}{2}\right)^s \cdot r}\right) = \mathcal{O}\left(rn^\omega + \frac{n^3}{r}\right)$$

Setze r so, dass die beiden Summanden gleich sind, also $r = n^{\frac{3-\omega}{2}}$. Damit ergibt sich für apsd die Laufzeit $\mathcal{O}\left(n^{\frac{3+\omega}{2}}\right)$.

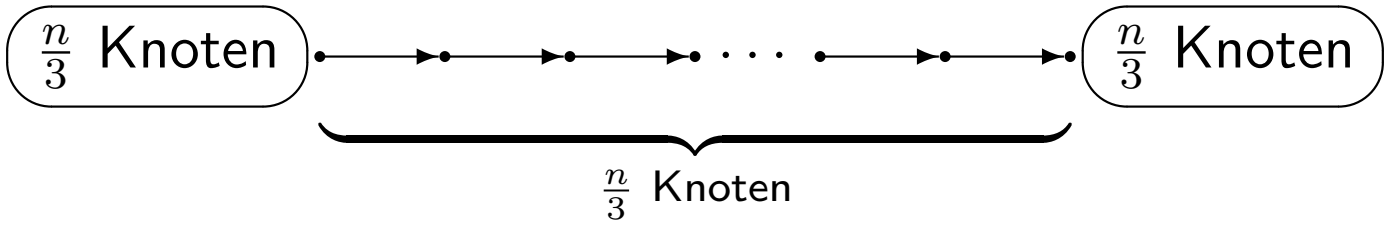
Satz 119

Das all-pairs-shortest-distance-Problem für Digraphen mit Kantenlänge $= 1$ lässt sich in Zeit $\mathcal{O}(n^{\frac{3+\omega}{2}})$ lösen (ω Exponent für Matrixmultiplikation).

Beweis:

$\surd$ □

Bemerkung: Beim apsp kann die Größe der Ausgabe $\Omega(n^3)$ sein:

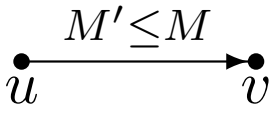


Verwende stattdessen die kürzeste-Pfade-Nachfolger-Matrix $N \in [0, \ldots, n-1]^{n \times n}$, wo n_{ij} die Nummer des zweiten Knoten auf „dem“ kürzesten Pfad von Knoten i zu Knoten j ist.
Eine solche Matrixdarstellung für die kürzesten Pfade zwischen allen Knotenpaaren kann in Zeit $\mathcal{O}(n^{\frac{3+\omega}{2}} \cdot \log^c n)$ (für ein geeignetes $c > 0$) bestimmt werden.

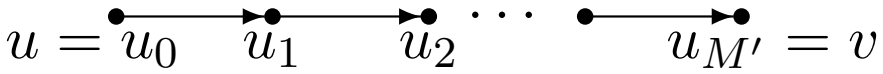
Satz 120

Für Digraphen mit Kantenlängen $\in \{1, 2, \ldots, M\}, M \in \mathbb{N}$, *kann APSD in Zeit* $\mathcal{O}((Mn)^{\frac{3+\omega}{2}})$ *gelöst werden.*

Beweis:

Idee: Ersetze 

durch:



□

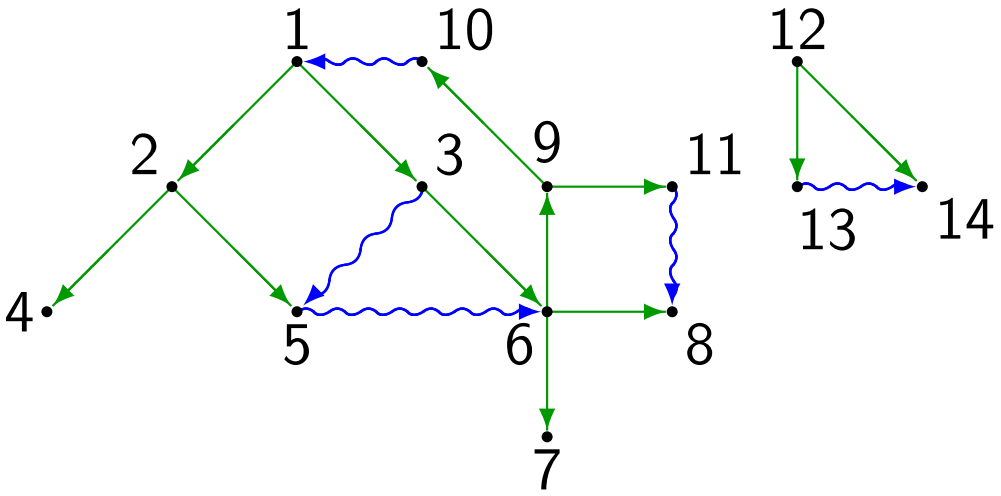
8. Ein Algorithmus für die transitive Hülle in Digraphen mit linearer erwarteter Laufzeit

Wir nehmen an, dass die Wahrscheinlichkeit eines Digraphen mit n Knoten und m Kanten (nur) eine Funktion von n und m ist.
Daraus folgt (wir lassen der Einfachheit halber Schleifen ($=$ Kanten $v \to v$) zu), dass jede Kante (u, v) mit Wahrscheinlichkeit $\frac{m}{n^2}$ vorhanden ist, falls wir Digraphen mit n Knoten und m Kanten betrachten.

Erinnerung: Breitensuche (BFS): Schlange, queue, FIFO:

Durchlaufe Graphen, indem wir, solange möglich, den vordersten Knoten v aus der Queue nehmen, ihn behandeln und die Liste seiner noch nicht behandelten Nachbarn in die Queue hinten einfügen.

Beispiel 121



Bezeichnungen:

Baumkanten

Querkanten

algorithm exp-lin-transitive-closure

0. Konstruiere die linear geordneten Adjazenzlisten L_i^r, $i = 1, \ldots, n$ des Graphen G^r (entsteht aus G durch Umkehrung aller Kanten).
   Beispiel:

$$\begin{array}{lcccc|lcccc} L_1 & = & 4 & 1 & 2 & L_1^r & = & 1 & 2 & 4 \\ L_2 & = & 1 & 4 & 3 & L_2^r & = & 1 & 3 & 4 \\ L_3 & = & 3 & 2 & & L_3^r & = & 2 & 3 & \\ L_4 & = & 1 & 4 & 2 & L_4^r & = & 1 & 2 & 4 \end{array}$$

   Ersetze ebenfalls alle L_i durch L_i^{rr} ($\rightarrow$ sortierte Adjazenzlisten)

1. Berechne für jeden Knoten i in BFS-Art eine Liste S_i von von i aus erreichbaren Knoten, so dass (i) oder (ii) gilt:
   (i) $|S_i| < \lfloor \frac{n}{2} \rfloor + 1$ und S_i enthält alle von i aus erreichbaren Knoten
   (ii) $|S_i| = \lfloor \frac{n}{2} \rfloor + 1$
2. Entsprechende Listen P_i von Vorgängern von i
3. **for** $i := 1$ **to** n **do**
   $L_i^* := S_i \cup \{j; i \in P_j\} \cup \{j;\ |S_i| = |P_j| = \lfloor \frac{n}{2} \rfloor + 1\}$
   od
   Bilde $(L_i^*)^{rr}$ für $i = 1, \ldots, n$

Korrektheit: $j \in L_i^*$ gdw (i,j) in transitiver Hülle.

Satz 122

Sei die Wahrscheinlichkeit eines Graphen (lediglich) eine Funktion der Anzahl n seiner Knoten und der Anzahl m seiner Kanten. Dann ist der Erwartungswert für die Laufzeit des obigen Algorithmus von Schnorr $\mathcal{O}(n + m^*)$. Dabei ist m^* der Erwartungswert für die Anzahl der Kanten in der transitiven Hülle.

Beweis:

Das Durchlaufen des Graphen (von v_1 aus für die Konstruktion von S_1) in BFS-Manier liefert eine Folge $(a_t)_{t\geq 1}$ von Knoten. Sei $L_{\sigma(\nu)}$ die ν-te Adjazenzliste, die in der BFS erkundet wird,

$$L_{\sigma(\nu)} = (a_t;\ h(\nu) < t \leq h(\nu+1))$$

Sei $\Delta L_{\sigma(\nu)}$ die Menge der a_t in $L_{\sigma(\nu)}$, und sei

$$S_i(t) := \{a_1, a_2, \ldots, a_t\}.$$

Wie bereits gezeigt, ist

$$\Pr[j \in L_i] = \frac{m}{n^2}, \quad \forall i, j$$

Beweis (Forts.):

Alle n-Tupel von geordneten Listen $L_1, L_2, \ldots, L_n$ mit

$$m = \sum_{i=1}^{n} |L_i|$$

sind aufgrund der Voraussetzungen über die Wahrscheinlichkeitsverteilung gleich wahrscheinlich. Also:

Beobachtung 1: Die Mengen $\Delta L_{\sigma(\nu)}$, $\nu = 1, 2, \ldots$ sind (paarweise) unabhängig.

Beobachtung 2: Die $\Delta L_{\sigma(\nu)}$ sind gleichverteilt, d.h. für alle $A \subseteq \{1, \ldots, n\}$ gilt:

$$\Pr[\Delta L_{\sigma(\nu)} = A] = (\frac{m}{n^2})^{|A|}(1 - \frac{m}{n^2})^{n-|A|}$$

Beweis (Forts.):

Lemma 123

Sei $A \subseteq \{1, \ldots, n\}$ mit $|A| = k$. Sei $(\bar{a}_t)_{t \geq 1}$ eine Folge von unabhängigen gleichverteilten Zufallsvariablen mit Wertemenge $\{1, \ldots, n\}$. Dann gilt:

$$\min\{t; \bar{a}_t \notin A\} \text{ hat Erwartungswert } 1 + \frac{k}{n-k}.$$

Beweis:

[des Lemmas] $\Pr[\bar{a}_1, \ldots, \bar{a}_r \in A \text{ und } \bar{a}_{r+1} \notin A] = \left(\frac{k}{n}\right)^r \cdot \left(1 - \frac{k}{n}\right)$.
Also ist der Erwartungswert für $\min\{t; \bar{a}_t \notin A\}$:

$$1 + \sum_{r=0}^{\infty} r \left(\frac{k}{n}\right)^r \left(1 - \frac{k}{n}\right) = 1 + \frac{k}{n}\left(1 - \frac{k}{n}\right) \sum_{r \geq 1} r \left(\frac{k}{n}\right)^{r-1}$$

$$= 1 + \frac{k}{n}\left(1 - \frac{k}{n}\right) \frac{1}{\left(1 - \frac{k}{n}\right)^2}$$

$$= 1 + \frac{\frac{k}{n}}{1 - \frac{k}{n}} = 1 + \frac{k}{n-k}\,.$$

□

Anmerkung:

$$\sum_{r=0}^{\infty} r z^{r-1} = \frac{d}{dz} \frac{1}{1-z} = \frac{1}{(1-z)^2}$$

Lemma 124

Sei $(\bar{a}_t)_{t\geq 1}$ *wie oben,* $k \leq \frac{n}{2}$*. Dann hat* $\min\{t; |\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_t| > k\}$ *Erwartungswert* $\leq \frac{3}{2}(k+1)$*.*

Beweis:

[des Lemmas] Wegen der Additivität des Erwartungswertes gilt: Der gesuchte Erwartungswert ist

$$\leq \sum_{\nu=0}^{k} \left(1 + \frac{\nu}{n-\nu}\right)$$

$$\leq k + 1 + \sum_{\nu=1}^{k} \frac{\nu}{\frac{n}{2}}$$

$$\leq k + 1 + \frac{k(k+1)}{n} \leq \frac{3}{2}(k+1)\,.$$

□

Beweis (Forts.):

Wenn wir jedes $L_{\sigma(\nu)}$ in sich zufällig permutieren, erhalten wir eine Folge $(\bar{a}_t)'_{t\geq 1}$ von Zufallsvariablen, so dass

$$|\{a_1,\ldots,a_t\}| = |\{a'_1,\ldots,a'_t\}| \text{ für } t = h(\nu), \quad \forall \nu$$

Da die a'_t innerhalb eines jeden Intervalls $h(\nu) < t \leq h(\nu+1)$ paarweise verschieden sind, gilt für sie die vorhergehende Abschätzung erst recht. Daher sind $(|S_1(t)|)_{t\geq 1}$ und $(|S_1(t)'|)_{t\geq 1}$ gleichverteilte Folgen von Zufallsvariablen, denn dies gilt zunächst für alle Zeitpunkte der Form $t = h(y)$; da aber für diese Zeitpunkte $S_1(t) = S'_1(t)$ ist und $h(\ )$ zufällig verteilt ist, ist auch die Verteilung der beiden Folgen zwischen diesen Intervallgrenzen identisch.

Beweis (Forts.):

Sei s der Erwartungswert und sei $|S_i|$ die tatsächliche Größe von S_i nach der Phase 1. Dann

$$s = \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor + 1} k \cdot \Pr[|S_i| = k] \,.$$

Die erwartete Anzahl der Schritte (und damit die Kosten) in der BFS sei q. Dann gilt:

$$q \leq \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor + 1} \Pr[|S_i| = k] \frac{3}{2} k = \frac{3}{2} s.$$

Also ist der Aufwand des Algorithmus für die Phasen 1 und 2 im Erwartungswert $\leq 3sn$. Da $ns = m^*$, und da die Kosten der anderen Phasen offensichtlich durch $\mathcal{O}(n + m + m^*)$ beschränkt sind, folgt die Behauptung. □

C.P. Schnorr:

An algorithm for transitive closure with linear expected time

SIAM J. Comput. **7**(2), pp. 127–133 (1978)