PS Algorithmen für verteilte Systeme

`https://avs.cs.sbg.ac.at/`

Aufgabenblatt 1

Abgabe bis Mittwoch, 18.03.2020, 11:00 Uhr auf `https://abgaben.cosy.sbg.ac.at/`

Aufgabe 1

Geben Sie eine Konfiguration der IDs in einem synchronen, nicht-anonymen, Ring mit n Knoten an, für die der Clockwise Algorithmus $\Theta(n)$ Nachrichten versendet und eine, für die der Clockwise Algorithmus $\Theta(n^2)$ Nachrichten versendet.

Aufgabe 2

Gegeben sei ein synchroner, anonymer, uniformer Ring mit n Knoten, in dem bereits ein Leader bestimmt wurde. Zeigen Sie, dass jedem Knoten in $O(n)$ Runden eine eindeutige ID zugeordnet werden kann.

PS Algorithmen für verteilte Systeme

https://avs.cs.sbg.ac.at/

Aufgabenblatt 2

Abgabe bis Mittwoch, 25.03.2020, 11:00 Uhr auf https://abgaben.cosy.sbg.ac.at/

Aufgabe 3

Gegeben sei ein synchrones, anonymes, uniformes Netzwerk mit n Knoten, dessen Graph ein Baum (also zusammenhängend und kreisfrei) ist. Zeigen Sie, dass ein Leader durch einen randomisierten Algorithmus mit in Erwartung $O(n)$ Runden und $O(n)$ Nachrichten bestimmt werden kann.

Hinweis: Der Baum hat keine designierte Wurzel. Versuchen Sie das randomisierte Tie-Breaking möglichst einfach zu gestalten.

Aufgabe 4

Gegeben sei ein synchroner, anonymer, non-uniformer Ring mit n Knoten (in dem also n globales Wissen ist). Außerdem sei eine Teilmenge L der Knoten mit $|L| \geq 1$ gegeben, wobei in Bezug auf diese Menge jeder Knoten initial nur weiß, ob er selbst zu L gehört oder nicht. Zeigen Sie, dass in $O(n)$ Runden und mit $O(n)$ Nachrichten bestimmt werden kann, ob es sich um eine gültige Menge von Leadern mit $|L| = 1$ handelt. Das Ziel ist, dass am Ende alle Knoten wissen, ob das der Fall ist oder nicht.

Anmerkung: Der gesuchte Verifikationsalgorithmus kann – mit ähnlichen Garantien – auch für asynchrone Ringe formuliert werden. In synchronen Ringen kann zusätzlich noch der Fall $|L| = 0$ erkannt werden.

Bonusaufgabe 1

Zeigen Sie – mit möglichst elementaren Methoden – dass $(1 - \frac{1}{n})^{n-1} \geq \frac{1}{e}$ für $n \geq 2$ gilt (wobei e die Eulersche Zahl ist). Diese Schranke ist relevant, wenn im randomisierten Leader-Election-Algorithmus IDs aus dem Bereich von 0 bis $n - 1$ vergeben werden.

Aufgabe 3: Leader Election im Baum

Extremfall: 

Idee: Münzwurf für beide Knoten -> 0 oder 1
Falls beide Knoten 0 oder beide Knoten 1: wiederhole
Ansonsten wird Knoten mit Ergebnis 1 zum Leader

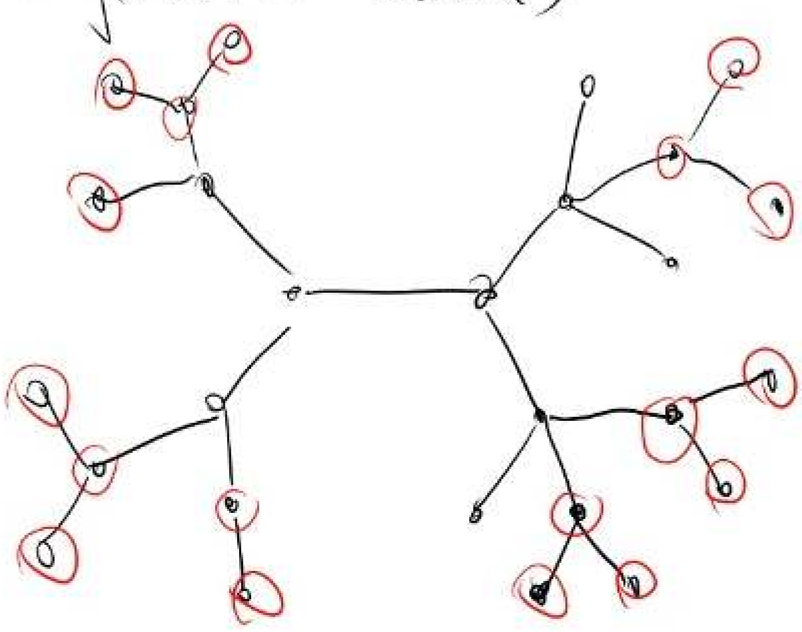
Erfolg: ein Knoten wird zum Leader
Erfolgswahrscheinlichkeit $q = \frac{1}{2}$

Erfolg: ein Knoten wird zum Leader
Erfolgswahrscheinlichkeit $q = \frac{1}{2}$

„Waiting Time Bound": In Erwartung $2 \left(= \frac{1}{q}\right)$ Wiederholungen bis zum ersten Erfolg

=> $O(1)$ Runden in Erwartung

Idee (für allgemeine Bäume)



- Blätter können sich als solche erkennen (#Nachbarn = 1)
- Knoten wird zum Follower, wenn all seine Kinder Follower geworden sind (#Nachbarn, die Follower sind = #Nachbarn − 1)
  -> Informiert Elternknoten (der noch kein Follower ist) darüber
- Ausnahme: Wenn in der Runde nach der Entscheidung zum Follower noch ein Nachbar zum Follower wird, dann starte Tie-Breaking mit diesem Nachbar

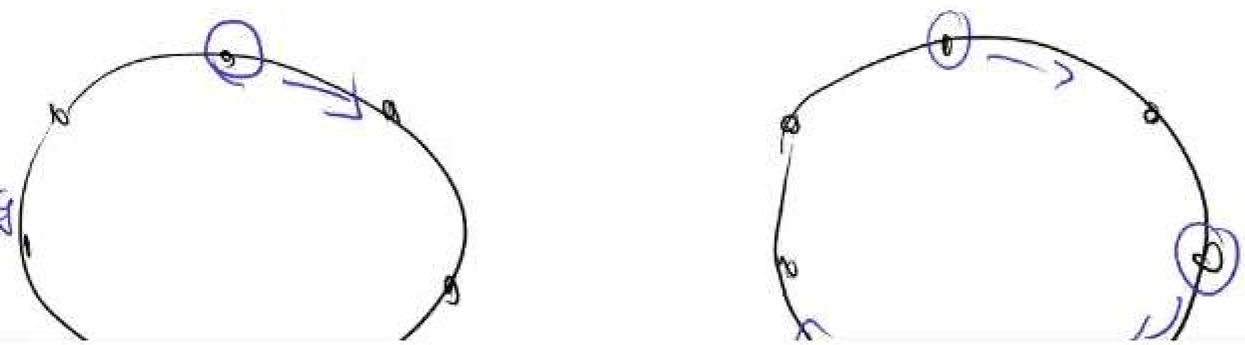
#Runden: $O(n) + O(1)$ in Erwartung
$O(n)$ in Erwartung

Nachrichten: $O(n) + O(1)$
$= O(n)$ in Erwartung

Synchron:

- Jeder Knoten in L sendet in erster Runde eine Nachricht im UZS
- Jeder Knoten $\notin L$ sendet jede empfangene Nachricht weiter im UZS
- Knoten $\in L$ leiten keine Nachrichten weiter
- In Runde $n+1$: Falls Knoten $\in L$ eine Nachricht empfängt, lautet das Ergebnis „gültig", ansonsten „ungültig", sende Nachricht mit Ergebnis im UZS

Beweisidee: Fallunterscheidung



Nachrichtenkomplexität: $O(n)$ (≤ 2 Nachrichten pro Knoten)

Bitkomplexität: $O(1)$ (Es reicht aus Nachrichten bestehend aus einem Bit zu versenden)

Asynchron: Idee: von Knoten in L initiierten Nachrichten werden zusätzlich mit einem „hop counter" versehen. (Anzahl an Weiterleitungen der Nachricht)

Dadurch können Knoten $\in L$ erkennen, ob empfangene Nachricht selbst initiiert wurde

PS Algorithmen für verteilte Systeme

`https://avs.cs.sbg.ac.at/`

Aufgabenblatt 3

Abgabe bis Mittwoch, 01.04.2020, 11:00 Uhr auf `https://abgaben.cosy.sbg.ac.at/`

Aufgabe 5

Zeigen Sie, dass im CONGEST Modell für ein beliebiges Netzwerk ein Leader in $O(D)$ Runden bestimmt werden kann, wobei D der Durchmesser des Netzwerks ist.

Aufgabe 6

Zeigen Sie, dass im CONGEST Modell der Durchmesser eines Netzwerks in $O(m)$ Runden berechnet werden kann, wobei m die Anzahl der Kanten des Netzwerks bezeichnet. Sie dürfen davon ausgehen, dass bereits ein Leader bestimmt wurde.

Hinweis: Gesucht ist eine generische Lösung, die für eine Vielzahl von Problemen funktioniert; die Berechnung des Durchmessers ist nur ein Beispiel von vielen. Im LOCAL Modell wären $O(D)$ Runden ausreichend.

Bonusaufgabe 2

In der Vorlesung haben wir den Queuing-Algorithmus für multiplen Upcast kennengelernt. Die Queue ist jedoch nicht unbedingt notwendig. Zeigen Sie, dass im CONGEST Modell für ein Netzwerk mit n Knoten und Durchmesser D ein multipler Upcast von k im Netzwerk verteilten Informationen mit jeweils Größe $O(\log n)$ an einen Wurzelknoten s über seinen gegebenen Breitensuchbaum in $O(k+D)$ Runden mit insgesamt $O(kD)$ Nachrichten durchgeführt werden kann, wenn jeder Knoten (zusätzlich zu den initial gespeicherten Informationen) nur Speicher proportional zur Anzahl seiner Nachbarn im Baum zur Verfügung hat.

Hinweis: Entwickeln Sie zunächst einen Algorithmus für den Spezialfall, dass sich alle Informationen an den Blättern des Breitensuchbaums befinden.

Aufgabe 5:

Für jeden Knoten v

Runde 1: Setze $T_v = ID(v)$

Sende $ID(v)$ an alle Nachbarn

Runde ≥ 2: Sei k die Anzahl in dieser Runde empfangenen Nachrichten

Falls $k \geq 1$

Bestimme die kleinste der empfangenen Nachrichten

Falls $M < T_v$

Setze $T_v = M$

Sende M an alle Nachbarn

Runde n zusätzlich:

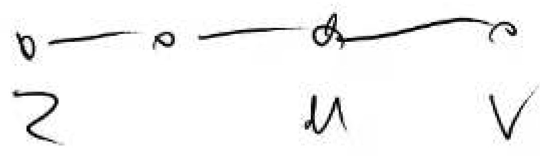
Falls $T_v = ID(v)$

v wird zum Leader

Ansonsten:

v wird zum Follower

Induktionshypothese: Nach Runde r gilt: $T_v = ID(z)$ für jeden Knoten v mit $dist(z,v) \leq r-1$, wobei z der Knoten mit kleinster ID ist



Korrektheit: Am Ende von Runde n gilt $T_v = ID(z)$ für jeden Knoten, weil $dist(z,v) \leq n-1$

$\downarrow$

$T_v = ID(v)$ nur $v = z$

$\Rightarrow$ Nur z wird zum Leader

Offensichtlich: $O(n)$ Runden

Nach $O(D)$ Runden ist $ID(z)$ allen Knoten bekannt

Laufzeit $O(D)$:

- Jeder Knoten v startet eine Breitensuchbaumberechnung mit Bestätigungsnachrichten (Convergecast), wobei jeder Nachricht in dieser Berechnung die ID von v angehängt wird.
- Jeder speichert die kleinste bisher gesehene ID und nimmt nur noch an Breitensuche des Knotens mit kleinster gesehener ID teil (Nachrichten von Breitensuchen mit höherer ID werden ignoriert)
- Sobald ein Knoten Bestätigungsnachrichten über Ende der Breitensuche von allen Kindern erhalten hat, wird er zum Leader, und informiert alle anderen Knoten über Downcast im Baum darüber (diese werden dann zu Followern)

Korrektheit:
- Breitensuche des Knotens mit kleinster ID setzt sich durch (jeder Knoten wird entsprechende Nachrichten immer weiter leiten)
- Für keinen anderen Knoten terminiert Breitensuche, weil Knoten mit kleinster ID keine Bestätigung sendet

Bandbreite: $O(\log n)$, weil jeder in jeder Runde nur an einer Breitensuche teilnimmt

Laufzeit: $O(D)$, weil Breitensuche des Leaders und Downcast $O(D)$ Runden benötigen

#Nachrichten: $O(Dm)$

Aufgabe 6: (Durchmesser $O(m)$ Runden)

- Leader berechnet Breitensuchbaum $O(D) = O(m)$ $D \le n-1 \le m$

  $m \ge n-1$, $(m \ge \frac{n}{2})$
- Jeder Knoten kodiert jede Kante zu einem Nachbar als Paar von IDs $(\_,\_) \leftarrow O(\log n)$ Bits
- Diese Kanteninformationen werden mit Algorithmus für multiplen Upcast an Leader gesendet $k = 2m$

  $O(k + D) = O(m)$
- Leader intern: rekonstruiert das Netzwerk aus Kanteninformationen das Netzwerk und berechnet den Durchmesser } keine Komm
- Leader Durchmesser $\hookrightarrow O(\log n)$ per Downcast an alle Knoten des Netzwerks $O(D)$

PS Algorithmen für verteilte Systeme

`https://avs.cs.sbg.ac.at/`

Aufgabenblatt 4

Abgabe bis Mittwoch, 22.04.2020, 11:00 Uhr auf `https://abgaben.cosy.sbg.ac.at/`

Aufgabe 7

Zeigen Sie, dass im CONGEST Modell ein Maximal Independent Set in $O(D)$ Runden mit einem deterministischen Algorithmus bestimmt werden kann, wenn das Netzwerk ein Baum ist, wobei D den Durchmesser des Netzwerks bezeichnet.

Aufgabe 8

Gegeben sei folgender Algorithmus um einen Leader in einem synchronen, anonymen, non-uniformen Ring zu bestimmen:

1. Setze $L = V$
2. Knoten aus L erhalten ID 0 mit Wahrscheinlichkeit p und ansonsten ID 1
3. Knoten aus $V \setminus L$ erhalten ID 2
4. Führe mit diesen IDs Clockwise Algorithmus (mit Präferenz für kleinere IDs) aus
5. Setze L auf die Menge der vom Clockwise Algorithmus bestimmten Leader
6. Falls $|L| > 1$, wiederhole ab Schritt 2

Argumentieren Sie, dass – für eine geeignete Wahl einer Konstanten $p < 1$ (z.B. $p = \frac{1}{3}$) – dieser Algorithmus so implementiert werden kann, dass er mit hoher Wahrscheinlichkeit $O(n \log n)$ Runden benötigt und $O(n \log n)$ Nachrichten versendet. Sie dürfen den Algorithmus aus Aufgabe 4 als „Black Box" verwenden.

Hinweis: Sie müssen insbesondere argumentieren, warum die allgemeine Schranke von $O(n^2)$ Nachrichten für den Clockwise Algorithmus aus der Vorlesung in dieser Anwendung zu pessimistisch ist.

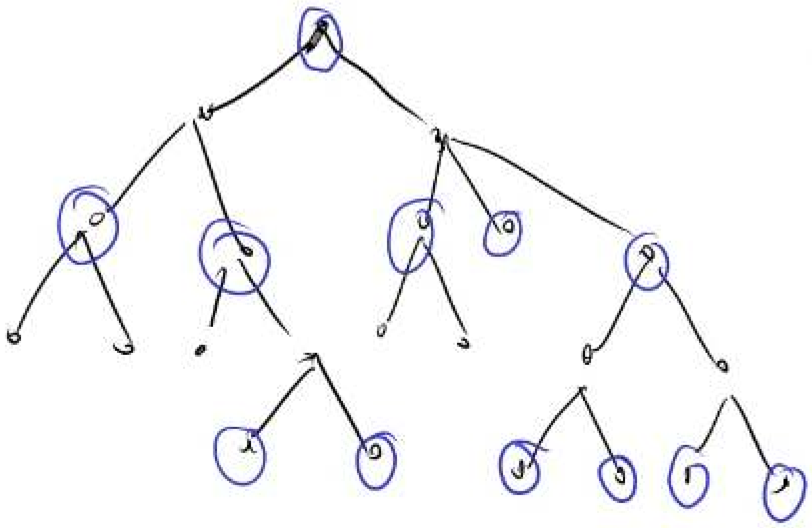
Anmerkung: Der Algorithmus kann – mit ähnlichen Garantien – auch für asynchrone Ringe formuliert werden. In synchronen Ringen kann darüber hinaus argumentiert werden, dass jede gesendete Nachricht nur aus konstant vielen Bits besteht. Dadurch erhält man eine Schranke auf die *Gesamtgröße* aller gesendeten Nachrichten, die um einen Faktor von $\log n$ niedriger ist als die des Radius Growth Algorithmus, während die Rundenzahl um einen Faktor $\log n$ höher ist.

Aufgabe 7 (MIS im Baum)

Bestimme zunächst Leader s in $O(D)$ Runden (siehe Aufgabe 5)

Behandle s als Wurzelknoten im Baum

Idee:



$U = \{v \in V \mid dist(s,v) \equiv 0 \mod 2\}$

ist MIS

$\rightarrow$ U ist ein Independent Set (ansonsten Kreis im Baum ↯)

$\rightarrow$ U ist maximal

<u>Idee</u>: Es müssen genügend viele „gute" Iterationen auftreten, die $|L|$ je um einen konstanten Faktor reduzieren, wie bei MIS Algorithmus

Betrachte eine Iteration mit anfangs $|L|$ Leader-„Kandidaten"

Sei N Zufallsvariable für die Anzahl der Knoten, die ID 0 erhalten

Für $v \in L$: $\Pr[v \text{ erhält ID } 0] = p$

In Iteration wird neue Menge L' an Leader-Kandidaten bestimmt

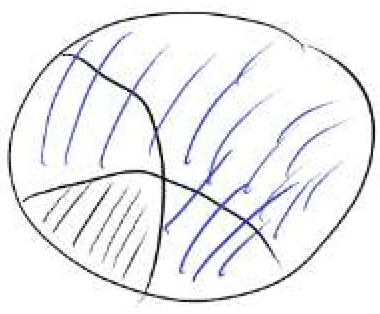
Es gilt: $$|L'| = \begin{cases} N & \text{falls } N \geq 1 \\ |L| & \text{ansonsten} \end{cases}$$

Iteration ist „gut", wenn $|L'| < \frac{2}{3}|L|$

Äquivalent: Iteration ist gut, wenn: 1. $N < \frac{2}{3}|L|$
2. $N \geq 1$

$\Pr[\text{Iteration gut}] = \Pr[N < \frac{2}{3}|L| \land N \geq 1] = 1 - \Pr[N \geq \frac{2}{3}|L| \lor N = 0]$

(Gegenereignis, deMorgan-Regeln)



$\Pr[N \geq \frac{2}{3}|L| \lor N = 0] \leq \Pr[N \geq \frac{2}{3}|L|] + \Pr[N = 0]$ (Union Bound)

$E[N] = p \cdot |L|$ (Erwartungswert Binomialverteilung)

$\Pr[N \geq \frac{2}{3}|L|] = \Pr[N \geq \frac{2}{3p} \cdot p|L|] = \Pr[N \geq \frac{2}{3p} \cdot E[N]] \leq \frac{3}{2}p$ (Markov Bound)

$$\Pr[N = 0] \underset{\text{Binomialverteilung}}{=} (1-p)^{|L|} \underset{|L| \geq 2}{\leq} (1-p)^2 = 1 - 2p + p^2$$

$\Rightarrow \Pr[N \geq \frac{2}{3}|L| \lor N = 0] \leq \frac{3}{2}p + 1 - 2p + p^2 = 1 + p^2 - \frac{p}{2}$

$$\Pr[\text{Iteration gut}] \geq 1 - (1 + p^2 - \frac{p}{2}) = \frac{p}{2} - p^2 \underset{p = \frac{1}{4}}{=} \frac{1}{8} - \frac{1}{16} = \frac{1}{16}$$

Jede gute ~~Phase~~ Iterat. reduziert $|L|$ um Faktor $\frac{2}{3}$

Nach k guten ~~Phase~~ Iterat. $|L| \leq (\frac{2}{3})^k \cdot n$

Für $k > \log_{\frac{2}{3}} \frac{1}{n}$ gilt: $|L| \leq (\frac{2}{3})^k \cdot n \leq (\frac{2}{3})^{\log_{\frac{2}{3}} \frac{1}{n}} \cdot n = \frac{1}{n} \cdot n = 1$

Sei Z ZV für #guter Iteration bei insgesamt l Iterationen

(Z ist bin. verteilt mit Einzelerfolgswahrsch. $q \geq \frac{1}{16}$)

$\mu := E[Z] \geq \frac{1}{16} l$

Setze $l = \lceil \frac{2}{\delta^2} \cdot 16 c \ln n \rceil$ für zu wählendes $\delta \in [0,1]$

$$\Pr[Z \leq \log_{\frac{2}{3}} \frac{1}{n}] = \Pr[Z \leq \frac{\log_{\frac{2}{3}} \frac{1}{n}}{\mu} \cdot \mu] \leq \Pr[Z \leq (1-\delta) \cdot \mu]$$

Stelle sicher, dass $\frac{\log_{\frac{2}{3}} \frac{1}{n}}{\mu} \leq (1-\delta)$

$$\leq \frac{1}{e^{\frac{\delta^2}{2} \cdot \mu}} \leq \frac{1}{e^{\frac{\delta^2}{2} \cdot \frac{1}{16} \cdot l}} \leq \frac{1}{e^{c \ln n}} = \frac{1}{n^c}$$

$\log_{\frac{2}{3}} \frac{1}{n} = \frac{\ln \frac{1}{n}}{\ln \frac{2}{3}} = \frac{\ln n}{\ln \frac{3}{2}} = 2.46\ldots \ln \leq 3 \cdot \ln$

$$> (1-\delta) \cdot \mu \geq (1-\delta) \frac{1}{16} \cdot l \geq (1-\delta) \cdot \frac{2}{\delta^2} \ln n \underset{\delta := \frac{1}{2}}{=} 4 \ln n > \log_{\frac{2}{3}} \frac{1}{n}$$

$\Rightarrow l = O(c \ln n)$ Iterationen d. Alg. sind mit Wahrsch. $\geq 1 - \frac{1}{n^c}$ ausreichend, um Leader zu bestimmen.

In jeder Iteration benötigt Clockwise Algorithmus mit 3 möglichen IDs $O(n)$ Runden und insgesamt $\leq 3n = O(n)$ Nachrichten, da jeder Knoten jede der drei IDs höchstens $1\times$ weiterleitet.

PS Algorithmen für verteilte Systeme

`https://avs.cs.sbg.ac.at/`

Aufgabenblatt 5

Abgabe bis Mittwoch, 29.04.2020, 11:00 Uhr auf `https://abgaben.cosy.sbg.ac.at/`

Aufgabe 9

In der Vorlesung haben wir nur Spanner-Konstruktionen für ungerichtete Graphen kennengelernt. Zeigen Sie, dass für *gerichtete* Graphen im Allgemeinen keine nicht-trivialen Spanner existieren, das heißt, dass es für jedes n einen gerichteten Graph mit n Knoten gibt, in dem jeder t-Spanner für $t < n$ mindestens $\Omega(n^2)$ Kanten hat.

Hinweis: Sie müssen nicht davon ausgehen, dass der Ausgangsgraph stark zusammenhängend ist, d. h., es darf Knoten u und v geben, für die es keinen Pfad von u nach v gibt (also $\text{dist}(u, v) = \infty$*).*

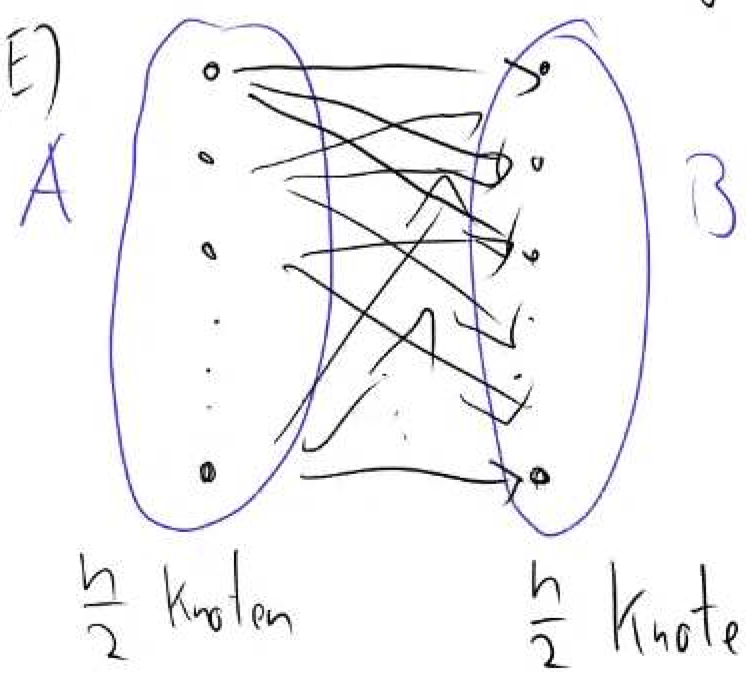
Aufgabe 10

In der Vorlesung haben wir einen Greedy-Algorithmus kennengelernt, der für eine beliebig gewählte Ganzzahl $k \geq 2$ einen $(2k-1)$-Spanner eines gegebenen ungerichteten, ungewichteten Graphen berechnet. Zeigen Sie, dass mit einer Modifikation des Greedy-Algorithmus für jeden gegebenen ungerichteten, *gewichteten* Graph ein $(2k-1)$-Spanner mit $O(n^{1+1/k})$ Kanten konstruiert werden kann.

Aufgabe 9 (Spanner für gerichtete Graphen?)

Allgemeine Konstruktion für n gerade:

$G=(V,E)$



- Jeder Knoten aus A hat eine Kante zu jedem Knoten aus B, aber zu keinem Knoten aus A
- Kein Knoten aus B hat ausgehende Kanten

$E = A \times B \quad |E| = \frac{n}{2} \cdot \frac{n}{2} = \frac{n^2}{4} = \Omega(n^2)$

Sei H ein echter Subgraph von G

$\Rightarrow$ Es gibt eine Kante (u,v), die in G ist aber nicht in H

Wegen Def. vom G: $u \in A$, $v \in B$

Beobachtung: In H gibt es keinen Pfad von u nach v

(Weil: jeder von u ausgehende Pfad besteht zunächst aus einer Kante (u,v') mit $v' \in B$ $v' \neq v$ und endet dann in v', weil v' keine ausgehenden Kanten hat)

$\Rightarrow dist_H(u,v) = \infty$ obwohl $dist_G(u,v) = 1$

$\Rightarrow$ Es gibt kein $t \in \mathbb{N}$ so dass $dist_H(u,v) \leq t \cdot dist_G(u,v)$

$\Rightarrow$ H hat unendlichen Stretch

$\Rightarrow$ G ist der einzige Spanner von G mit endlichem Stretch

$\Rightarrow$ Jeder Spanner von G hat $\Omega(n^2)$ Kanten

mit endlichem Stretch

Aufgabe 10 (Spanner für gewichtete Graphen)

Gegeben: $G = (V, E)$

Sortiere E aufsteigend nach Gewicht

$F \leftarrow \emptyset$

foreach $(u,v) \in E$ (in sortierter Reihenfolge) do

  Sei $H = (V, F)$

  if $\mathrm{dist}_H(u,v) > (2k-1) \cdot w_G(u,v)$ then

    $F \leftarrow F \cup \{(u,v)\}$

Lemma: $H = (V, F)$ ist $(2k-1)$-Spanner von G

Beweis Sei (u,v) beliebige Kante von G

Falls $(u,v) \in F$: $\mathrm{dist}_H(u,v) \leq w_H(u,v) = w_G(u,v) \leq (2k-1) \cdot w_G(u,v)$

Falls $(u,v) \notin F$:

Sei $H' = (V, F')$ der Zustand von H direkt vor der Entscheidung „gegen" (u,v)

Da $F' \subseteq F$: $\mathrm{dist}_H(u,v) \leq \mathrm{dist}_{H'}(u,v) \leq (2k-1) \cdot w_G(u,v)$

Lemma: H hat ungewichteten Girth $> 2k$

Beweis: Annahme: H hat einen Kreis K bestehend $\leq 2k$ Kanten

Sei (u,v) die Kante von K, die im Algorithmus zuletzt hinzugefügt wurde

Vor dem Hinzufügen: Kein Weg von u nach v der Länge $\leq (2k-1) \cdot w_G(u,v)$ in H

$\Rightarrow w_H(K \setminus \{u,v\}) > (2k-1) \cdot w_G(u,v)$

Andererseits: Jede Kante auf K hat wegen Sortierung Gewicht $\leq w_G(u,v)$

$\Rightarrow w_H(K \setminus \{u,v\}) \leq (2k-1) \cdot w_G(u,v)$ ↯

$\Rightarrow H$ hat keinen Kreis der Länge $\leq 2k$

Aus VO: Jeder Graph mit Girth $> 2k$ hat $O(n^{1+1/k})$ Kanten

$\Rightarrow H$ ist ein $(2k-1)$-Spanner von G mit $O(n^{1+1/k})$ Kanten

PS Algorithmen für verteilte Systeme

https://avs.cs.sbg.ac.at/

Aufgabenblatt 6

Abgabe bis Mittwoch, 06.05.2020, 11:00 Uhr auf https://abgaben.cosy.sbg.ac.at/

Aufgabe 11

Zeigen Sie, dass im CONGEST Modell für jedes Netzwerk $G = (V, E)$ mit n Knoten aus einem gegebenen (ρ, μ, ℓ)-Cover in konstant vielen Runden ein $(2\rho + 1)$-Spanner von G mit $O(\mu + \ell n)$ Kanten konstruiert werden kann, wenn jeder Knoten für jedes Cluster, in dem er enthalten ist, seinen Parent und seine Kinder im Baum des Clusters sowie die IDs des Clusterzentrums kennt.

Aufgabe 12

Zeigen Sie, dass folgender Algorithmus einen $(2k - 1)$-Spanner $H = (V, F)$ mit $O(n^{1+1/k})$ Kanten für einen Graph $G = (V, E)$ mit n Knoten berechnet:

1. $F = \emptyset$
2. Wähle beliebigen Knoten $s \in V$
3. Berechne Breitensuchbaum T von s in $G = (V, E)$ sowie $L_i(s) = \{v \in V \mid \mathrm{dist}_G(s, v) = i\}$ für alle $i \geq 0$
4. Sei $i(s)$ das kleinste i so dass $|L_i(s)| \leq |L_{i-1}(s)| \cdot n^{1/k}$
5. Füge für jeden Knoten in $L_0(s) \cup \cdots \cup L_{i(s)}(s)$ die Kante zum Elternknoten in T zu F hinzu
6. Entferne alle Knoten in $L_0(s) \cup \cdots \cup L_{i(s)-1}(s)$ aus V und entferne alle Kanten mit Endpunkten in $L_0(s) \cup \cdots \cup L_{i(s)-1}(s)$ aus E
7. Falls $V \neq \emptyset$, Wiederhole ab Schritt 2

Anmerkung: Die Ungleichung $|L_i(s)| \leq |L_{i-1}(s)| \cdot n^{1/k}$ kann als Abbruchbedingung in der Durchführung der Breitensuche verwendet werden. Dadurch kann der Algorithmus im sequentiellen RAM Modell in linearer Zeit implementiert werden.

Bonusaufgabe 3

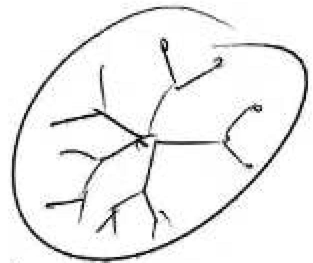
Zeigen Sie, dass in der asynchronen Variante des CONGEST Modells ein Breitensuchbaum in $O(D^2)$ Runden mit $O(m + nD)$ Nachrichten konstruiert werden kann, wobei n die Anzahl der Knoten, m die Anzahl der Kanten und den D den Durchmesser des Netzwerks bezeichnet.

Aufgabe 11 (Cover-Spanner)

Sei (ρ, μ, ℓ) (bestehend aus Clusterings $\mathcal{C}_1, \dots, \mathcal{C}_\ell$)

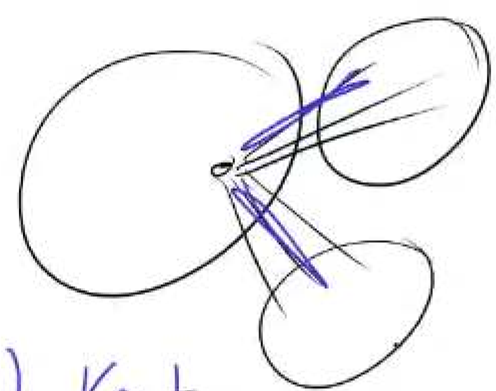
Konstruiere folgenden Spanner H:

- Für jeden Knoten v und jedes Cluster, in dem v enthalten ist, füge Kante zum Parent im Cluster des Baums zu Spanner hinzu



Jeder Knoten in $\leq \ell$ Clusterings/Cluster enthalten
$\Rightarrow \leq n \cdot \ell$ solcher Kanten

- Für jedes $1 \leq i \leq \ell$ und jeden Knoten v in Clustering $\mathcal{C}_i$, füge für jedes Nachbarcluster in $\mathcal{C}_i, \dots, \mathcal{C}_\ell$ eine Kante von v nach C zu H hinzu



$\leq \mu$ Kanten (nach Def. des Covers)

$\Rightarrow$ H hat $O(\ell n + \mu)$ Kanten.

Stretch: Sei (u,v) beliebige Kante aus G

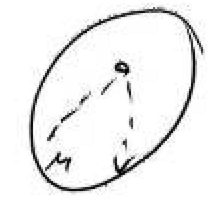
Sei i max. Index so dass u in Cluster aus $\mathcal{C}_i$ enthalten

Sei j min. –––––"–– v ––"––––– $\mathcal{C}_j$ –"–

OBdA $i \leq j$

Falls u und v im selben Cluster: Pfad von u nach v der Länge $\leq 2\rho$ über Cluster-Zentrum

$\leq 2\rho + 1$



Falls u und v in verschiedenen Clustern:

Da Cluster von v zu u benachbart ist (wg Existenz der Kante (u,v)), gibt es eine Kante $(u,v') \in H$ von in das Cluster von v

$\Rightarrow$ Pfad der Länge $\leq 2\rho + 1$ von u nach v in H

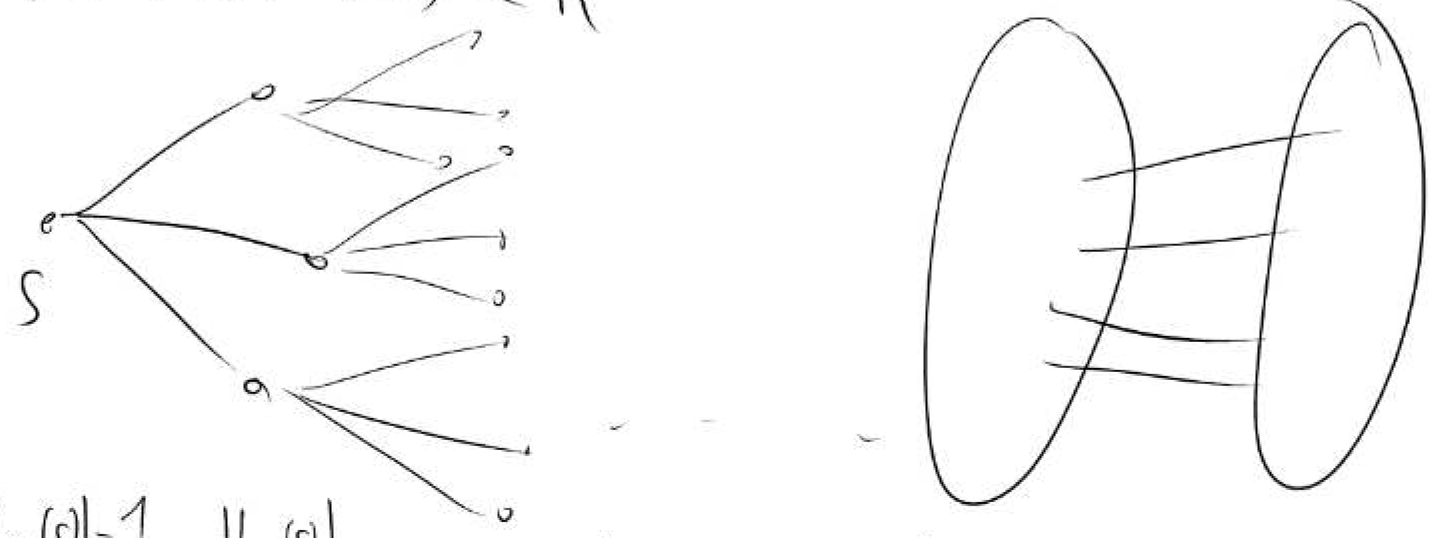
$((u,v') - (v' - \text{Cluster-Zentrum}) - (\text{Cluster-Zentrum} - v))$

Aufgabe 12 (Linearzeit Spanner)

Sei S die Menge der Startknoten, von denen aus Breitensuchbäume berechnet wurden

Für jedes $s \in S$, sei $i(s)$ das kleinste i, so dass $|L_i(s)| \leq |L_{i-1}(s)| \cdot n^{1/k}$

Beobachtung: $\forall s \in S: i(s) \leq k$



$|L_0(s)| = 1$ $\quad |L_1(s)| > n^{1/k}$ $\quad |L_2(s)| > (n^{1/k})^2 = n^{2/k}$ $\quad |L_{k-1}(s)| > (n^{1/k})^{k-1}$ $\quad |L_k(s)| > (n^{1/k})^k = n$ ✓

Lemma: H hat Stretch $2k-1$

Beweis: Sei (u,v) beliebige Kante aus G

Kante (u,v) wird im Algorithmus dann entfernt, wenn

entweder $u \in L_0(s) \cup \ldots \cup L_{i(s)-1}(s)$ oder

$v \in L_0(s) \cup \ldots \cup L_{i(s)-1}(s)$ (für ein $s \in S$)

OBdA $u \in L_0(s) \cup \ldots \cup L_{i(s)-1}(s)$

$\Rightarrow v \in L_0(s) \cup \ldots \cup L_{i(s)}(s)$

$\Rightarrow$ Breitensuchbaum von s (und somit H) enthält Pfad von u nach v (über s) der Länge

$\leq i(s) - 1 + i(s) = 2\,i(s) - 1 \leq 2k-1$

Lemma: H hat $O(n^{1+1/k})$ viele Kanten $(|F| = O(n^{1+1/k}))$

Beweis

$$|F| = \sum_{s \in S} \sum_{1 \leq i \leq i(s)} |L_i(s)| = \sum_{s \in S} \sum_{1 \leq i \leq i(s)-1} |L_i(s)| + \sum_{s \in S} |L_{i(s)}(s)|$$

disjunkt!

disjunkt

$$\leq n + \sum_{s \in S} |L_{i(s)-1}(s)| \cdot n^{1/k} \leq n + n \cdot n^{1/k} \leq 2 \cdot n^{1+1/k}$$

PS Algorithmen für verteilte Systeme

`https://avs.cs.sbg.ac.at/`

Aufgabenblatt 8

Abgabe bis Mittwoch, 13.05.2020, 11:00 Uhr auf `https://abgaben.cosy.sbg.ac.at/`

Aufgabe 13

Zeigen Sie, dass im CONGEST Modell eine 2-Approximation $\hat{D}$ des Durchmessers D des Netzwerks in $O(D)$ Runden bestimmt werden kann. Gesucht ist also eine Zahl $\hat{D}$, so dass $\frac{1}{2}D \leq \hat{D} \leq D$. Sie dürfen davon ausgehen, dass bereits ein Leader bestimmt wurde.

Hinweis: Die Dreiecksungleichung besagt, dass $\mathrm{dist}(u, v) \leq \mathrm{dist}(u, w) + \mathrm{dist}(w, v)$ für alle Knoten u, v und w.

Aufgabe 14

Zeigen Sie, dass im CONGEST Modell eine $O(\log n)$-Approximation des APSP Problems für ungewichtete Graphen mit n Knoten durch einen Las-Vegas Algorithmus mit erwarteter Rundenzahl $O(n \log n)$ berechnet werden kann. Jeder Knoten u soll also am Ende für jeden anderen Knoten v eine Zahl $\delta(u, v)$ kennen, so dass $\mathrm{dist}(u, v) \leq \delta(u, v) \leq O(\log n) \cdot \mathrm{dist}(u, v)$. (Sie können davon ausgehen, dass bereits ein Leader bestimmt wurde.)

Hinweis: In der Vorlesung wurde nur ein Monte Carlo-*Algorithmus für die* exakte *APSP-Berechnung vorgestellt. Es gibt mehrere Lösungsansätze für diese Aufgabe, insbesondere wäre es möglich den Durchmesser in der gefordeten erwarteten Laufzeit auch exakt zu berechnen.*

Aufgabe 13 (Approximation Durchmesser)

Sei s der Leader $\rightarrow O(D)$

Berechnen Breitensuchbaum mit Distanzen von s aus $O(D)$

Setze $\hat{D} = Ecc(s)$ $O(D)$ aggr. Upcast

$$Ecc(s) = \max_{v \in V} dist(s,v)$$

$\hat{D} \leq D$, da $\hat{D} = Ecc(s) = \max\limits_{v \in V} dist(s,v) \leq \max\limits_{u \in V} \max\limits_{v \in V} dist(u,v) = D$

Zu zeigen: $\frac{1}{2} \cdot D \leq \hat{D} \Leftrightarrow D \leq 2\hat{D}$

Seien $u, v \in V$ beliebig

Dann gilt nach Δ-Ungleichung:

$$\begin{aligned} dist(u,v) &\leq dist(u,s) + dist(s,v) \\ &= dist(s,u) + dist(s,v) \\ &\leq Ecc(s) + Ecc(s) = 2 \cdot Ecc(s) = 2 \cdot \hat{D} \end{aligned}$$

$D = \max\limits_{u,v \in V} dist(u,v) \leq 2\hat{D}$

Aufgabe 14: (APSP Las Vegas)

Methode 1:

- Berechne α-Spanner $H=(V,F)$
- Mache Spanner H allen Knoten bekannt
- Berechne APSP auf H intern

$k = \lceil \log n \rceil$ Baswana-Sen: $O(k^2) = O(n \log n)$

$(2\lceil \log n - 1)$-Spanner mit in Erw. $|F| = O(n \log n)$ Kanten

$\hookrightarrow O(|V| + |F| + D) = O(n \log n)$ in Erwartung

$\text{dist}_G(u,v) \leq \text{dist}_H(u,v) \leq \alpha \cdot \text{dist}_G(u,v)$ für alle Knotenpaare u, v

$\Rightarrow$ Alg. berechnet α-Approx. für APSP

PS Algorithmen für verteilte Systeme

https://avs.cs.sbg.ac.at/

Aufgabenblatt 8

Abgabe bis Mittwoch, 20.05.2020, 11:00 Uhr auf https://abgaben.cosy.sbg.ac.at/

Aufgabe 15

Gegeben sei ein (Monte-Carlo) Algorithmus für das (exakte) SSSP Problem für gewichtete Graphen, der im CONGEST Modell mit hoher Wahrscheinlichkeit eine korrekte Lösung berechnet (und mit geringer Wahrscheinlichkeit falsche Distanzwerte berechnet). Zeigen Sie, dass die Korrektheit einer Ausgabe dieses Algorithmus (bestehend aus einer Distanz für jeden Knoten) in $O(D)$ Runden verifiziert werden kann.

Hinweis: Man kann eine lokale „Optimalitätsbedingung“ angeben.

Aufgabe 16

Zeigen Sie, dass im CONGEST Modell eine $(1 + \epsilon)$-Approximation für das SSSP Problem für gewichtete Graphen mit n Knoten in $O(\sqrt{nD} \log^2(n)/\epsilon)$ Runden berechnet werden kann (wenn das höchste Kantengewicht W polynomiell in n ist, also $W = n^{o(1)}$.

Hinweis: „Simulieren“ Sie eine geeignete Variante von Dijkstras Algorithmus auf dem Overlay Netzwerk $H = (Z, Z \times Z)$, dessen Knoten die Zentren sind und dessen Kantengewichte den approximativen h-Distanzen zwischen den Zentren entsprechen.

Aufgabe 15 (SSSP Verifikation)

Gegeben: Distanzschätzung $\delta(s,v)$ für jeden Knoten v

Lemma: Falls $\delta(s,s)=0$ und $\delta(s,v)=\min_{(u,v)\in E}(\delta(s,u)+w(u,v))$ für alle $v\neq s$

dann gilt: $\delta(s,v)=\text{dist}(s,v)$ für alle Knoten v

Falls Lemma bewiesen, funktioniert folgender Alg:
- $\delta(s,v)$ wird in einer Runde allen Nachbarn bekannt gemacht (für alle $v\in V$)
- Jeder Knoten prüft intern Optimalitätsbedingung des Lemmas
- In $O(D)$ Runden wird bestimmt, ob Bedingung für alle zutrifft

Falls ja: akzeptiere Distanzschätzung. Falls nein: lehne Distanzschätzung ab

Beweis des Lemmas: Zeige: $\delta(s,v)\leq \text{dist}(s,v)$ und $\delta(s,v)\geq \text{dist}(s,v)$
-> Gleichheit

Beobachtung: Für jeden Knoten v gibt es einen Pfad von s nach v der Länge $\delta(s,v)$
→ Folge für jeden Knoten v der Kante (u,v) (bzw. einer dieser Kanten), für die $\delta(s,v)=\delta(s,u)+w(u,v)$ gilt

$\Rightarrow \delta(s,v)\geq \text{dist}(s,v)\ \forall v$

Sei T ein Baum kürzester Wege von s aus

Wir zeigen per Induktion über Tiefe der Knoten in T, dass $\delta(s,v)\leq \text{dist}(s,v)$

IB (Tiefe 0): $\delta(s,s)=0=\text{dist}(s,s)$

IS (Tiefe $i\geq 1$): Sei u' Elternknoten von v in T

$$\delta(s,v)=\min_{(u,v)\in E}(\delta(s,u)+w(u,v))\leq \delta(s,u')+w(u',v)$$

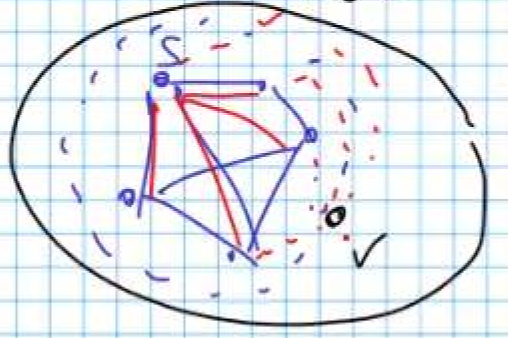
Da u' Tiefe $i-1$ hat gilt wegen IH: $\delta(s,u')\leq \text{dist}(s,u')$

Somit: $\delta(s,v)\leq \text{dist}(s,u')+w(u',v)=\text{dist}(s,v)$

$\Rightarrow \delta(s,v)\leq \text{dist}(s,v)$

Aufgabe 16 (Dijkstra auf Overlay)

Alg: · Führe Schritte 1 und 2 aus wie in der VO (Sampling, approx. h-Distanzen)
· Berechne SSSP von s im Graph $H = (Z, Z \times Z)$ mit Gewichte
$w_H(x,y) = \tilde{d}(x,y)$ für alle $x, y \in Z$
und mache $d_H(s,x)$ für jeden $x \in Z$ allen Knoten des Netzwerks bekannt
· Für jeden Knoten v: Berechne intern
$$\delta(s,v) = \min_{x \in Z} \left(d_H(s,x) + \tilde{d}(x,v) \right) \text{ als Ergebnis}$$



Lemma: $\delta(s,v) = d_{H_v}(s,v)$ wobei $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$
mit Gewichten $w_{H_v}(x,y) = \tilde{d}(x,y)$

Beweis: Wegen Dreiecksungl. gilt für jedes $x \in Z$:
$$\begin{aligned} d_{H_v}(s,v) &\leq d_{H_v}(s,x) + d_{H_v}(x,v) \\ &\overset{H \subseteq H_v}{\leq} d_H(s,x) + d_{H_v}(x,v) \\ &\leq d_H(s,x) + w_{H_v}(x,v) \\ &= d_H(s,x) + \tilde{d}(x,v) \end{aligned}$$
Somit: $d_{H_v}(s,v) \leq \min_{x \in Z} (d_H(s,x) + \tilde{d}(x,v)) = \delta(s,v)$

Sei $x \in Z$ der Vorgängerknoten auf kürzestem Weg π von s nach v in H_v
$$\Rightarrow d_{H_v}(s,v) = \underbrace{d_{H_v}(s,x)}_{\text{Länge eines Teilpfads von } \pi \text{, der nur Knoten aus Z beinhaltet}} + \underbrace{w_{H_v}(x,v)}_{\tilde{d}(x,v) \text{ nach Def. von } H_v}$$
$$\begin{aligned} &\geq d_H(s,x) + \tilde{d}(x,v) \\ &\geq \min_{x' \in Z} (d_H(s,x') + \tilde{d}(x',v)) = \delta(s,v) \end{aligned}$$
$\square$

Dijkstra:
$d(s) = 0$
$d(x) = \infty$ für alle $x \in Z \setminus \{s\}$
$\mathcal{U} = \emptyset$
While $\mathcal{U} \neq Z$
  Setze x auf einen Knoten aus $Z \setminus \mathcal{U}$ mit minimalem $d(x)$
  $\mathcal{U} = \mathcal{U} \cup \{x\}$
  Für jede Kante (x,y) (für jedes $y \in Z_u$) setze $d(y) = \min(d(y), d(x) + w_H(x,y))$

Jede Iteration kann in $O(D)$ Runden durchgeführt werden
Somit: Laufzeit $O(D \cdot |Z|)$ für Dijkstra auf Overlay

Gesamtlaufzeit: $O((|Z| + h) \cdot \log(nW) \cdot \log(n)/\varepsilon + |Z| \cdot D)$
mit $|Z| = O(\frac{n}{h} \log n)$ $= O\left(\left(\frac{n}{h} \cdot \log n + h\right) \cdot \log(nW) \log(n)/\varepsilon + \frac{n}{h} \cdot \log n \cdot D \right)$

$D \cdot \frac{n}{h} = h \Leftrightarrow h^2 = nD \Leftrightarrow h = \sqrt{nD}$

$\Rightarrow O(\sqrt{nD} \log(nW) \cdot \log^2(n)/\varepsilon)$

PS Algorithmen für verteilte Systeme

`https://avs.cs.sbg.ac.at/`

Aufgabenblatt 9

Abgabe bis Mittwoch, 27.05.2020, 11:00 Uhr auf `https://abgaben.cosy.sbg.ac.at/`

Aufgabe 17

Gegeben sei ein Las-Vegas-Algorithmus für ein Problem P im CONGEST Modell mit einer (allen Knoten explizit bekannten) erwarteten Laufzeit von $R(n)$ Runden für ein Netzwerk mit n Knoten. Zeigen Sie, dass es für ein Netzwerk mit n Knoten und Durchmesser D einen Monte-Carlo-Algorithmus für P im CONGEST Modell gibt, der, für jedes gegebene $c \geq 1$, immer Laufzeit $O((R(n) + D) \cdot c \log n)$ hat und mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$ korrekt ist.

Hinweis: Markov Bound

Aufgabe 18

Gegeben sei ein beliebiges Entscheidungsproblem P (d.h. es gibt nur Ausgaben der Form YES oder NO). Angenommen, wir haben einen randomisierten Algorithmus $\mathscr{A}$ für P mit den folgenden Eigenschaften:

- Für alle Eingaben $x \in P$ gilt $\Pr[\mathscr{A}(x) = \mathsf{NO}] \leq 1/3$ und
- für alle Eingaben $x \notin P$ gilt $\Pr[\mathscr{A}(x) = \mathsf{YES}] \leq 1/3$.

Zeigen Sie, dass man durch logarithmisch viele Wiederholungen von $\mathscr{A}$ einen Algorithmus für das Problem P mit Fehlerwahrscheinlichkeit $1/n^c$ (für eine beliebige vorgegebene Konstante c) erhalten kann.

Hinweis: Chernoff Bound

Aufgabe 17 (Las Vegas → Monte Carlo)

Gegeben: Las-Vegas Alg. $\mathcal{A}$ für Problem P mit erwarteter Laufzeit $R(n)$ $(E[X] = R(n))$

X: Zufallsvariable für empirische Laufzeit

$Pr[X \geq 2R(n)] = Pr[X \geq 2 \cdot E[X]] \leq \frac{1}{2}$ (Markov Bound $\alpha = 2$)

$\Rightarrow Pr[X < 2R(n)] = 1 - Pr[X \geq 2R(n)] \geq \frac{1}{2}$

Monte-Carlo Algorithmus $\mathcal{B}$:

- Bestimme Leader ℓ ← $O(D)$
- Wiederhole höchstens $\lceil c \log n \rceil$

  Führe $\mathcal{A}$ für $2R(n)$ Runden aus ← $O(R(n))$

  Jeder Knoten informiert ℓ, ob $\mathcal{A}$ aus seiner Sicht terminiert ist ← $O(D)$ (Upcast mit Aggregation)

  Falls $\mathcal{A}$ für jeden Knoten terminiert ist, informiert ℓ, dass abgebrochen wird, ansonsten informiert ℓ, dass nächste Wiederholung stattfindet ← $O(D)$ (Downcast)

Interpretiere jede Wiederholung von $\mathcal{A}$ als Bernoulli-Experiment mit Erfolgswahrscheinlichkeit $p \geq \frac{1}{2}$

$P[\text{keine der } \lceil c \log n \rceil \text{ Wh. erfolgreich}] = (1-p)^{\lceil c \log n \rceil} \leq \frac{1}{2^{c \log n}} = \frac{1}{n^c}$

$P[\mathcal{B} \text{ erfolgreich}] \geq 1 - \frac{1}{n^c}$

Aufgabe 18 (Boosting der Erfolgswahrscheinlichkeit)

- Wiederhole Algorithmus $k = \lceil 48 c \ln n \rceil$ Mal
- Fälle Gesamtentscheidung für das Ergebnis, das in der Mehrheit der Wiederholungen ausgegeben wurde

Zufallsvariable X_i: $X_i = \begin{cases} 1 & \text{falls Alg. entscheidet in } i\text{-ter Wiederholung korrekt} \\ 0 & \text{andernfalls} \end{cases}$

$$\Pr[\text{Gesamtentscheidung korrekt}] = \Pr\left[\sum_{i=1}^{k} X_i > \frac{k}{2}\right]$$
$$= 1 - \underline{\Pr\left[\sum_{i=1}^{k} X_i \leq \frac{k}{2}\right]}$$

$$E\left[\sum_{i=1}^{k} X_i\right] = \sum_{i=1}^{k} E[X_i] = \sum_{i=1}^{k} \underbrace{\Pr[X_i = 1]}_{\geq \frac{2}{3}} \geq \sum_{i=1}^{k} \frac{2}{3} = \frac{2}{3} \cdot k$$

$$\Pr\left[\sum_{i=1}^{k} X_i \leq \frac{k}{2}\right] = \Pr\left[\sum_{i=1}^{k} X_i \leq \left(1 - \frac{1}{4}\right) \cdot \frac{2}{3} k\right]$$
$$\leq \Pr\left[\sum_{i=1}^{k} X_i \leq (1 - \underbrace{\tfrac{1}{4}}_{\delta}) \cdot \underbrace{E\left[\sum_{i=1}^{k} X_i\right]}_{\mu}\right]$$

Chernoff Bound

$$\leq \frac{1}{e^{\frac{\delta^2}{2} \cdot \mu}} = \frac{1}{e^{\frac{1}{32} \cdot \mu}} \underset{\mu \geq \frac{2}{3} k}{\leq} \frac{1}{e^{\frac{1}{32} \cdot \frac{2}{3} k}} = \frac{1}{e^{\frac{1}{48} \cdot k}}$$

$$\underset{k \geq 48 c \ln n}{\leq} \frac{1}{e^{c \ln n}} = \frac{1}{n^c}$$

$$\Pr[\text{Gesamtentscheidung korrekt}] = 1 - \Pr\left[\sum_{i=1}^{k} X_i \leq \frac{k}{2}\right]$$
$$\geq 1 - \frac{1}{n^c}.$$

PS Algorithmen für verteilte Systeme

https://avs.cs.sbg.ac.at/

Aufgabenblatt 10

Abgabe bis Mittwoch, 03.06.2020, 11:00 Uhr auf https://abgaben.cosy.sbg.ac.at/

Aufgabe 19

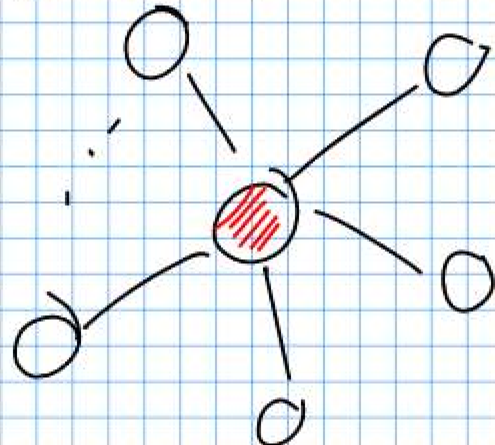
In der Vorlesung analysieren wir den Prozess der epidemischen Informationsausbreitung nur für vollständige Graphen. Dabei werden sowohl im Push- als auch im Pull-Modell mit hoher Wahrscheinlichkeit $\Theta(\log n)$ viele Runden benötigt bis alle n Knoten des Netzwerks infiziert sind. Zeigen Sie, anhand einer geeigneten Klasse von Beispielgraphen, dass sich für allgemeine Graphen mit n Knoten bei ungünstiger Startkonfiguration die Anzahl der benötigten Runden im Push- und Pull-Modell um *mehr* als einen konstanten Faktor unterscheiden kann.

Aufgabe 20

In der Vorlesung wurde bewiesen, dass während der Wachstumsphase im Push-Modell der Wachstumsfaktor für die Anzahl der infizierten Knoten mit Wahrscheinlichkeit höchstens $\frac{1}{e^{1/24}}$ höchstens $\frac{7}{6}$ beträgt, also dass $\Pr[I(t+1) \leq \frac{7}{6} I(t)] \leq \frac{1}{e^{1/24}}$ unter der Voraussetzung $I(t) \leq \frac{n}{3}$ gilt. Zeigen Sie, ausgehend von dieser Ungleichung, mittels Anwendung der Chernoff-Bound, dass die Wachstumsphase aus $O(\log n)$ Runden besteht.

Aufgabe 19 (Separation von Push&Pull auf allgemeinen Graphen)

Sterngraph mit n Knoten und infiziertem zentralen Knoten



Push Modell: • Jeder äußere Knoten kann nur vom zentralen Knoten infiziert werden
• Der zentrale Knoten infiziert höchstens einen Knoten pro Runde
$\Rightarrow$ mindestens $n-1$ Runden bis alle (äußeren) Knoten infiziert sind

Pull Modell: • Jeder Knoten ruft garantiert den zentralen Knoten an
• Daher werden in einer Runde bereits alle (äußeren) Knoten infiziert

Aufgabe 20 (Länge der Wachstumsphase im Push-Modell)

Def: Runde t ist gut, falls $I(t+1) > \frac{7}{6}\cdot I(t)$

Nach ~~höchstens~~ $> \log_{\frac{7}{6}} \frac{n}{3}$ guten Runden gilt $I(t) > \frac{1}{3} n$

$$\left(I(1)=1,\ \left(\tfrac{7}{6}\right)^k > \tfrac{1}{3} n \Rightarrow k > \log_{\frac{7}{6}} \tfrac{n}{3} \right)$$

Definiere ZV $X(t) = \begin{cases} 1 & \text{falls Runde } t \text{ gut} \\ 0 & \text{ansonsten} \end{cases}$

Gesucht: k so dass in k Runden mit hoher Wahrscheinlichkeit $> \log_{\frac{7}{6}} \frac{n}{3}$ gute Runden auftreten, also dass

$$\Pr\left[\sum_{t=1}^{k} X(t) > \log_{\frac{7}{6}} \frac{n}{3} \right] \geq 1 - \frac{1}{n^c}$$

Erwartungswert: $\Pr[X(t)=1] = 1 - \underbrace{\Pr[X(t)=0]}_{\leq \frac{1}{e^{1/24}} \text{ aus V0}} \geq \frac{4}{100}$

$$E\left[\sum_{t=1}^{k} X(t)\right] = \sum_{t=1}^{k} \Pr[X(t)=1] \geq k \cdot \frac{4}{100}$$

Chernoff Bound (Ziel: $\Pr\left[\sum_{t=1}^{k} X(t) \leq \log_{\frac{7}{6}} \frac{n}{3}\right] \leq n^c$

$$\Pr\left[\sum_{t=1}^{k} X(t) \leq (1-\delta) \cdot E\left[\sum_{t=1}^{k} X(t)\right] \right] \leq \frac{1}{e^{\frac{\delta^2}{2} \cdot E\left[\sum_{t=1}^{k} X(t)\right]}} \leq \frac{1}{e^{\frac{\delta^2}{2} \cdot k \cdot \frac{4}{100}}} \leq \frac{1}{n^c}$$

$$\log_{\frac{7}{6}} \frac{n}{3} \leq (1-\delta) \cdot \underbrace{E\left[\sum_{t=1}^{k} X(t)\right]}_{\substack{\geq \\ k \cdot \frac{4}{100}}}$$

$$k \cdot \frac{4}{100} \geq \frac{50 c \ln n}{\delta^2} \cdot \frac{4}{100}$$

$$\frac{\delta^2}{2} \cdot k \cdot \frac{4}{100} \geq c \ln n$$

$$\Leftrightarrow k \geq \frac{50 c \ln n}{\delta^2}$$

$$k = \left\lceil \frac{50 c \ln n}{\frac{1}{4}} \right\rceil = O(\log n)$$

Ziel: setze δ so, dass $\log_{\frac{7}{6}} \frac{n}{3} \leq (1-\delta) \cdot \frac{50 c \ln n}{\delta^2} \cdot \frac{4}{100}$

PS Algorithmen für verteilte Systeme

https://avs.cs.sbg.ac.at/

Aufgabenblatt 11

Abgabe bis Mittwoch, 10.06.2020, 11:00 Uhr auf https://abgaben.cosy.sbg.ac.at/

Aufgabe 21

Zeigen Sie, dass im Push-Modell für n Knoten folgendes gilt: Wenn $c' \ln n \leq G(t) \leq \frac{2}{3}n$ für eine passende Konstante c' gilt, dann ist $G(t+1) \leq 0.9 \cdot G(t)$ mit hoher Wahrscheinlichkeit (also mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$ für eine vorgegebene Konstante c).

Hinweis: Die Aussage gilt jedenfalls für $c' = 288c$.

Aufgabe 22

Zeigen Sie, dass im Pull-Modell für n Knoten, ausgehend von einem infizierten Knoten, nach $O((\log n)^2)$ Runden mit hoher Wahrscheinlichkeit mindestens $\frac{n}{\ln n}$ Knoten infiziert sind.

Anmerkung: Diese Aufgabe vervollständigt die Analyse des Pull-Modells, die wir in der Vorlesung nur für den Fall $I(t) > \frac{n}{\ln n}$ durchgeführt haben. Es gibt mehrere Wege, diese Aufgabe zu lösen, und insbesondere ist es mit einfachen Mitteln möglich, eine bessere Schranke von $O(\log n)$ zu zeigen. Der Faktor $\frac{1}{\ln n}$ ist für die Analyse nicht zentral; um die Notation möglichst einfach zu halten, kann es hilfreich sein, auf die etwas stärkere Garantie abzuzielen, mindestens $\frac{n}{3}$ Knoten zu infizieren.

Aufgabe 21 (Schrumpfen gesunder Knoten)

z.z. $G(t+1) \le 0.9\, G(t)$ falls $c' \ln n \le G(t) \le \frac{2}{3} n$

im Push-Model $\qquad I(t) \ge \frac{1}{3} n$

Wahrscheinlichkeit, dass ein fixer gesunder Knoten nicht infiziert wird

$$\left(1-\frac{1}{n}\right)^{I(t)} \le \left(1-\frac{1}{n}\right)^{\frac{1}{3} n} = \Big(\underbrace{\left(1-\frac{1}{n}\right)^{n}}_{\le \frac{1}{e}}\Big)^{\frac{1}{3}} \le \frac{1}{e^{1/3}} \approx 0{,}716\ldots$$

$\ge$ (da $I(t) \le n$)

$$\left(1-\frac{1}{n}\right)^{n} \ge \frac{1}{4}$$

$(1+\delta)\cdot \frac{1}{e^{1/3}} \le 0.9$

Wähle $\delta = \frac{1}{4}$

$$X_j(t) = \begin{cases} 1 & \text{gesunde Knoten } j \text{ wird nicht infiziert} \\ 0 & \text{ansonsten} \end{cases}$$

$$G(t+1) = \sum_{j=1}^{G(t)} X_j(t) \qquad E[G(t+1)] = \sum_{j=1}^{G(t)} \overbrace{\underbrace{E[X_j(t)]}_{\ge \frac{1}{4}}}^{\le \frac{1}{e^{1/3}}} \begin{array}{l} \le \frac{1}{e^{1/3}}\, G(t) \\ \ge \frac{1}{4}\, G(t) \end{array}$$

Mit $\delta = \frac{1}{4}$ gilt $(1+\delta)\cdot \frac{1}{e^{1/3}} \le 0.9$

Somit: $\Pr[G(t+1) \ge 0.9 \cdot G(t)]$

$$\le \Pr\left[G(t+1) \ge (1+\delta)\cdot \frac{1}{e^{1/3}}\, G(t)\right]$$

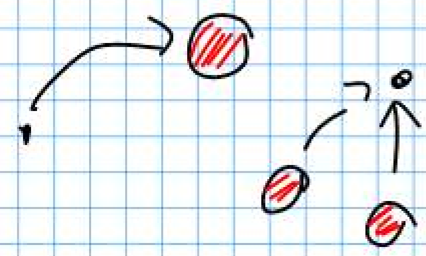
$$\le \Pr\Big[\underbrace{G(t+1)}_{=\sum_{j=1}^{G(t)} X_j} \ge (1+\delta)\cdot E[\underbrace{G(t+1)}_{\sum_{j=1}^{G(t)} X_j}]\Big]$$

$$\le \frac{1}{e^{\frac{\delta^2}{3}\cdot E[G(t+1)]}} \le \frac{1}{e^{\frac{\delta^2}{3}\frac{1}{4}\cdot G(t)}} = \frac{1}{e^{\frac{1}{192} G(t)}}$$

$\Rightarrow$ Mit $G(t) \ge 192 \cdot c \cdot \ln n$ gilt $\Pr[G(t+1) \ge 0.9\, G(t)]$

$$\le \frac{1}{e^{c \ln n}} = \frac{1}{n^c}$$

Aufgabe 22 (Startphase Pull Modell)



Annahme: $I(t) \leq \frac{1}{3}n \Rightarrow G(t) \geq \frac{2}{3}n$

$N(t)$: Anzahl neu infizierter Knoten

$I(t+1) = I(t) + N(t)$

$$X_j(t) = \begin{cases} 1 & \text{falls gesunder Knoten } j \text{ in Runde } t \text{ infiziert wird} \\ 0 & \text{ansonsten} \end{cases}$$

$$N(t) = \sum_{j=1}^{G(t)} X_j(t)$$

$$E[N(t)] = \sum_{j=1}^{G(t)} E[X_j(t)] = \sum_{j=1}^{G(t)} Pr[X_j(t) = 1] = \sum_{j=1}^{G(t)} \frac{I(t)}{n} = \sum_{j=1}^{G(t)} i(t)$$

$$= G(t) \cdot i(t) \geq \frac{2}{3}n \cdot i(t) = \frac{2}{3}n \cdot \frac{I(t)}{n} = \frac{2}{3} I(t)$$

$$Pr\left[N(t) \leq \frac{1}{3} I(t)\right] = Pr\left[N(t) \leq \left(1 - \frac{1}{2}\right) \cdot \frac{2}{3} I(t)\right] \leq Pr\left[N(t) \leq \left(1 - \frac{1}{2}\right) E[N(t)]\right]$$

$$\overset{\text{Chernoff Bound}}{\leq} \frac{1}{e^{\frac{1}{8} \cdot E[N(t)]}} \leq \frac{1}{e^{\frac{1}{12} \cdot I(t)}}$$

Wir haben gezeigt $Pr\left[N(t) \leq \frac{1}{3} I(t)\right] \leq \frac{1}{e^{\frac{1}{12} \cdot I(t)}} \leq \frac{1}{n^c}$ falls $I(t) \geq 12 c \ln n$

$\Rightarrow$ Wachstum um Faktor $\frac{4}{3}$ in jeder Runde

$\Rightarrow O(\log n)$ Runden bis $I(t) \geq \frac{1}{3}n$

Wahrscheinlichkeit, dass in $\frac{12 c \ln n}{I(t)}$ aufeinanderfolgenden Runden immer $N(t) \leq \frac{1}{3} I(t)$ gilt:

$$\leq \left(\frac{1}{e^{\frac{1}{12} I(t)}}\right)^{\frac{12 c \ln n}{I(t)}} \leq \left(\frac{1}{e}\right)^{c \ln n} = \frac{1}{n^c}$$

$\Rightarrow$ Mit hoher Wahrscheinlichkeit wächst $I(t)$ um Faktor $\frac{4}{3}$ nach $\frac{12 c \ln n}{I(t)}$ Runden (oder Faktor 2 nach $36 \frac{c \ln n}{I(t)}$ Runden)

$$\left(\frac{4}{3}\right)^3 \geq 2$$

Um $I(t) \geq 12 c \ln n$ werden mit hoher Wahrscheinlichkeit

$$\sum_{k=0}^{\log(12 c \ln n)} \frac{36 c \ln n}{2^k} = O(c \ln n)$$ Runden benötigt

2^k: Anzahl infizierter Knoten nach k Verdopplungen

PS Algorithmen für verteilte Systeme

https://avs.cs.sbg.ac.at/

Aufgabenblatt 12

Abgabe bis Mittwoch, 17.06.2020, 11:00 Uhr auf https://abgaben.cosy.sbg.ac.at/

Aufgabe 23

Schreiben Sie ein Programm, das den Push-Algorithmus auf einem vollständigen Graph mit n Knoten simuliert. Ihr Programm muss nicht verteilt oder parallel laufen, es soll den Algorithmus lediglich simulieren. Geben Sie für jedes $n \in \{10, 100, 1000, 10000\}$ den Mittelwert der Laufzeiten von 50 unabhängigen Simulationsdurchläufen aus, das heißt die durchschnittliche Anzahl der Runden, die benötigt wird, um alle Knoten zu infizieren (ausgehend von einem infizierten Knoten). Ermitteln Sie anschließend diejenige Konstante c, für die der empirisch gemessene Mittelwert der Laufzeiten (näherungsweise) als $c \ln n$ ausgedrückt werden kann. Bitte gestalten Sie Ihre Abgabe so, dass sie den Quelltext, die Simulationsergebnisse und Ihre Berechnungen enthält.

Hinweis: Achten Sie bei sequentieller Ausführung darauf, dass in jeder Runde jeder gesunde Knoten nur von denjenigen Knoten infiziert werden kann, die bereits am Anfang der Runde infiziert waren.

Aufgabe 24

Angenommen, eine Münze wird n Mal hintereinander geworfen. Sei, für jedes $1 \le i \le n$, X_i die binäre Zufallsvariable, die 1 ist, wenn der i-te Münzwurf Kopf zeigt, und ansonsten 0. Weiters sei $X := \sum_{i=1}^{n} X_i$ die Zufallsvariable, die angibt, wie oft die Münze bei n Würfen Kopf zeigt, und sei A das Ereignis, dass X seinen Erwartungswert um mindestens 5 Prozent übersteigt.

Schreiben Sie ein Computerprogramm in einer Programmiersprache Ihrer Wahl, das n Münzwürfe simuliert. Wiederholen Sie diese Simulation für jedes $n \in \{10, 100, 1000, 10000\}$ jeweils 50 Mal und bestimmen Sie für jedes n die relative Häufigkeit des Ereignisses A, also den relativen Anteil an Simulationswiederholungen, in denen X stark von seinem Erwartungswert abweicht. Vergleichen Sie anschließend für jedes $n \in \{10, 100, 1000, 10000\}$ (a) die Chernoff-Schranke für die Wahrscheinlichkeit des Ereignisses A mit (b) der empirisch ermittelten relativen Häufigkeit von A. Führen Sie Ihre Simulation mit zwei verschiedenen Zufallsgeneratoren durch, etwa `rnd` und `random.SystemRandom()` in Python oder `java.util.Random` und `java.security.SecureRandom` in Java, und erläutern Sie mit welchem Zufallsgenerator das Ereignis A häufiger auftritt. Bitte gestalten Sie Ihre Abgabe so, dass sie den Quelltext, die Simulationsergebnisse und Ihre Berechnungen enthält.

Bonusaufgabe 4

Zeigen Sie, dass es in einem *anyonymen* Ring unmöglich ist, ein Maximal Independent Set zu berechnen (auch nicht in einem synchronen, non-uniformen Ring).

Aufgabe 23

n	⌀ Rundenzahl	c
10	6.7	2.9
100	12.4	2.7
1000	18.0	2.6
10000	24.0	2.6

Triviale untere Schranke:

$$\geq \log_2 n = 1{,}44\ldots \ln n$$

$$\leadsto c \geq 1{,}44$$

Aufgabe 24

n-maliger Münzwurf

$$X_i = \begin{cases} 1 & \text{wenn } i\text{-ter Münzwurf Kopf} \\ 0 & \text{ansonsten} \end{cases}$$

$$X = \sum_{i=1}^{n} X_i \quad (= \text{Anzahl Kopf})$$

$$\mu := E[X] = \frac{n}{2}$$

$$\Pr[X \geq (1+0{,}05)\cdot\mu] \leq \frac{1}{e^{\frac{(0{,}05)^2}{3}\cdot\mu}} = \frac{1}{e^{n/2400}}$$

n	Chernoff	empirisch
10	0.996	≈ 0.38
100	0.959	≈ 0.31
1000	0.659	≈ 0.060
10.000	0.0155	≈ 0.00