Maximal Independent Set

Algorithmen für verteilte Systeme

Sebastian Forster

Universität Salzburg

Globale vs. lokale Probleme

Globale Probleme:

- Benötigen $\Omega(D)$ Runden
- Kürzeste Wege
- Minimum Spanning Tree

Globale vs. lokale Probleme

Globale Probleme:

- Benötigen $\Omega(D)$ Runden
- Kürzeste Wege
- Minimum Spanning Tree

Lokale Probleme:

- Laufzeit $o(n)$
- Maximal Independent Set
- Maximal Matching
- Spanner-Berechnung

Elementare Wahrscheinlichkeitstheorie

1. Union Bound
2. Linearität des Erwartungswerts
3. Markov Bound
4. Chernoff Bound

Union Bound und Erwartungswert

Lemma (Union Bound)

Seien A und B zwei Ereignisse. Dann gilt $\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$.

Union Bound und Erwartungswert

Lemma (Union Bound)

Seien A und B zwei Ereignisse. Dann gilt $\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$.

Allgemein: $\Pr[\bigcup_i A_i] \leq \sum_i \Pr[A_i]$

Union Bound und Erwartungswert

Lemma (Union Bound)

Seien A und B zwei Ereignisse. Dann gilt $\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$.

Allgemein: $\Pr[\bigcup_i A_i] \leq \sum_i \Pr[A_i]$

Definition

Der **Erwartungswert** einer Zufallsvariable X ist $\mathrm{Ex}[X] = \sum_x x \cdot \Pr[X = x]$.

Für binäre Zufallsvariablen $X \in \{0, 1\}$ gilt: $\mathrm{Ex}[X] = \Pr[X = 1]$.

Union Bound und Erwartungswert

Lemma (Union Bound)

Seien A und B zwei Ereignisse. Dann gilt $\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$.

Allgemein: $\Pr[\bigcup_i A_i] \leq \sum_i \Pr[A_i]$

Definition

Der **Erwartungswert** einer Zufallsvariable X ist $\mathrm{Ex}[X] = \sum_x x \cdot \Pr[X = x]$.

Für binäre Zufallsvariablen $X \in \{0, 1\}$ gilt: $\mathrm{Ex}[X] = \Pr[X = 1]$.

Lemma (Linearität des Erwartungswerts)

Seien X und Y zwei Zufallsvariablen. Dann gilt $\mathrm{Ex}[X + Y] = \mathrm{Ex}[X] + \mathrm{Ex}[Y]$

Union Bound und Erwartungswert

Lemma (Union Bound)

Seien A und B zwei Ereignisse. Dann gilt $\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$.

Allgemein: $\Pr[\bigcup_i A_i] \leq \sum_i \Pr[A_i]$

Definition

Der **Erwartungswert** einer Zufallsvariable X ist $\mathrm{Ex}[X] = \sum_x x \cdot \Pr[X = x]$.

Für binäre Zufallsvariablen $X \in \{0, 1\}$ gilt: $\mathrm{Ex}[X] = \Pr[X = 1]$.

Lemma (Linearität des Erwartungswerts)

Seien X und Y zwei Zufallsvariablen. Dann gilt $\mathrm{Ex}[X + Y] = \mathrm{Ex}[X] + \mathrm{Ex}[Y]$

Allgemein: $\mathrm{Ex}[\sum_i X_i] = \sum_i \mathrm{Ex}[X_i]$

Motivation Concentration Bounds

- Erwartungswert alleine oft nicht aussagekräftig genug

Motivation Concentration Bounds

- Erwartungswert alleine oft nicht aussagekräftig genug
- Wie groß ist Wahrscheinlichkeit vom Erwartungswert um multiplikativen Faktor abzuweichen?

Motivation Concentration Bounds

- Erwartungswert alleine oft nicht aussagekräftig genug
- Wie groß ist Wahrscheinlichkeit vom Erwartungswert um multiplikativen Faktor abzuweichen?
- Algorithmische Anwendung: Wie viele Wiederholungen nötig, um gewünschte Schranke für Wahrscheinlichkeit der Abweichung zu erhalten?

Markov Bound

Theorem (Markov Bound)

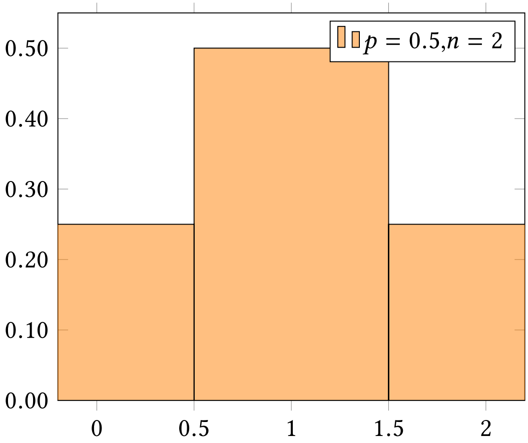
Sei X eine nicht-negative Zufallsvariable und $\alpha > 0$. Dann gilt

$$\Pr[X \geq \alpha \cdot \mathrm{Ex}[X] \leq \frac{1}{\alpha} \,.$$

Binomialverteilung

n Wiederholungen eines Bernoulli-Experiments:

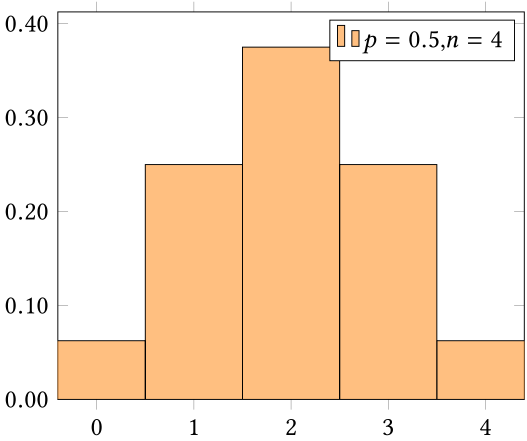
- $\Pr[X_i = 1] = p$
- $\Pr[X_i = 0] = 1 - p$
- Verteilung von $\sum_{i=1}^{n} X_i$:



Binomialverteilung

n Wiederholungen eines Bernoulli-Experiments:

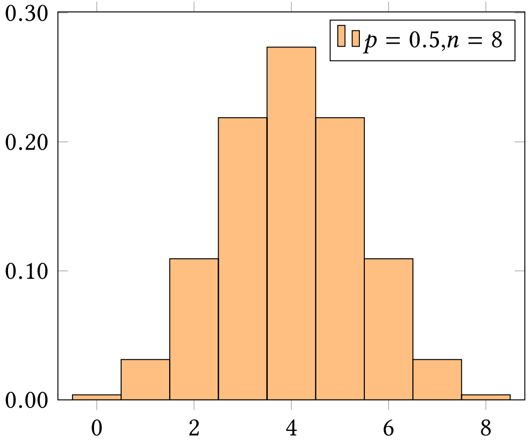
- $\Pr[X_i = 1] = p$
- $\Pr[X_i = 0] = 1 - p$
- Verteilung von $\sum_{i=1}^{n} X_i$:



Binomialverteilung

n Wiederholungen eines Bernoulli-Experiments:

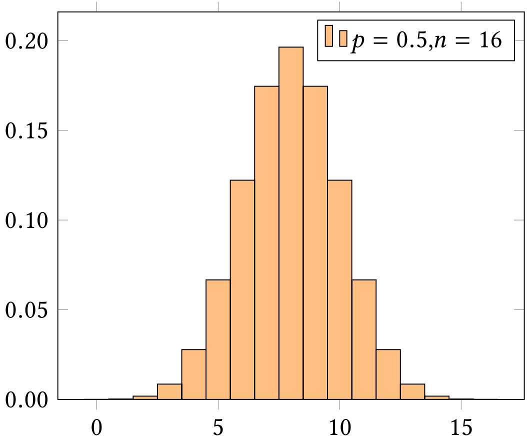
- $\Pr[X_i = 1] = p$
- $\Pr[X_i = 0] = 1 - p$
- Verteilung von $\sum_{i=1}^{n} X_i$:



Binomialverteilung

n Wiederholungen eines Bernoulli-Experiments:

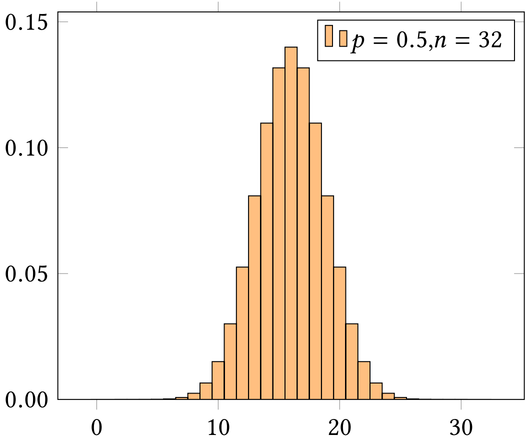
- $\Pr[X_i = 1] = p$
- $\Pr[X_i = 0] = 1 - p$
- Verteilung von $\sum_{i=1}^{n} X_i$:



Binomialverteilung

n Wiederholungen eines Bernoulli-Experiments:

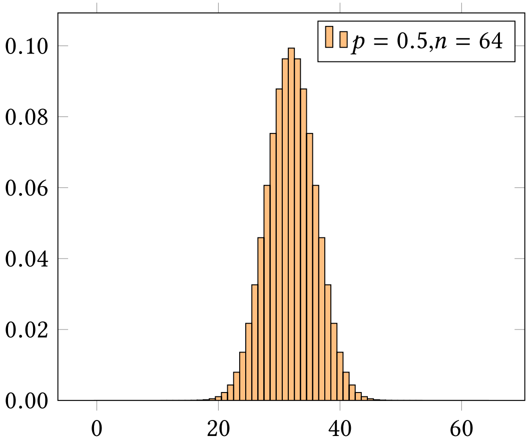
- $\Pr[X_i = 1] = p$
- $\Pr[X_i = 0] = 1 - p$
- Verteilung von $\sum_{i=1}^{n} X_i$:



Binomialverteilung

n Wiederholungen eines Bernoulli-Experiments:

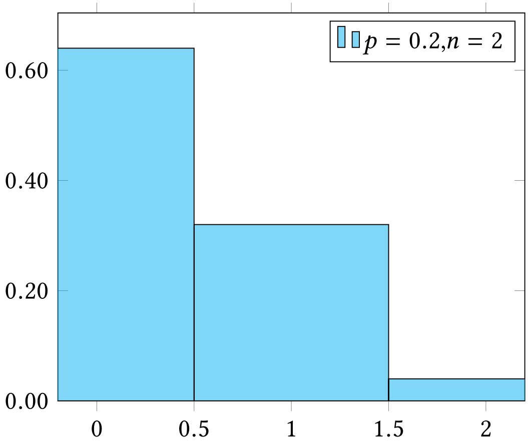
- $\Pr[X_i = 1] = p$
- $\Pr[X_i = 0] = 1 - p$
- Verteilung von $\sum_{i=1}^{n} X_i$:



Binomialverteilung

n Wiederholungen eines Bernoulli-Experiments:

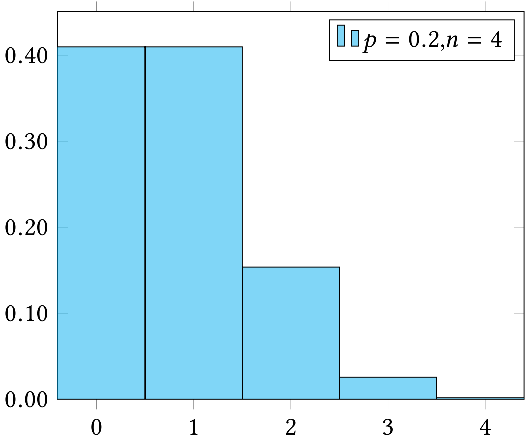
- $\Pr[X_i = 1] = p$
- $\Pr[X_i = 0] = 1 - p$
- Verteilung von $\sum_{i=1}^{n} X_i$:



Binomialverteilung

n Wiederholungen eines Bernoulli-Experiments:

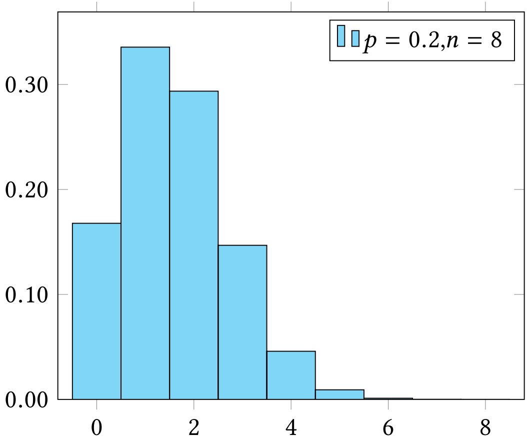
- $\Pr[X_i = 1] = p$
- $\Pr[X_i = 0] = 1 - p$
- Verteilung von $\sum_{i=1}^{n} X_i$:



Binomialverteilung

n Wiederholungen eines Bernoulli-Experiments:

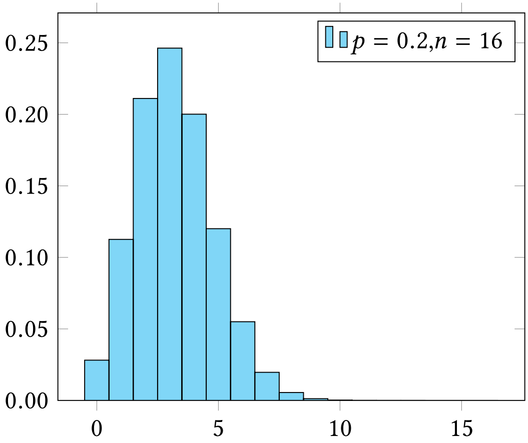
- $\Pr[X_i = 1] = p$
- $\Pr[X_i = 0] = 1 - p$
- Verteilung von $\sum_{i=1}^{n} X_i$:



Binomialverteilung

n Wiederholungen eines Bernoulli-Experiments:

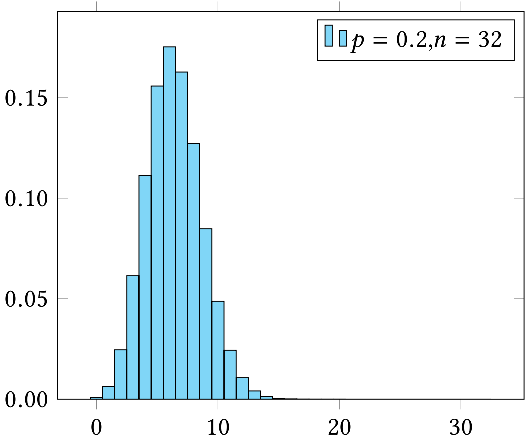
- $\Pr[X_i = 1] = p$
- $\Pr[X_i = 0] = 1 - p$
- Verteilung von $\sum_{i=1}^{n} X_i$:



Binomialverteilung

n Wiederholungen eines Bernoulli-Experiments:

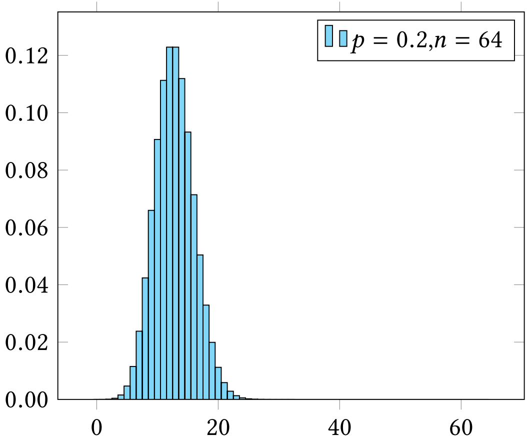
- $\Pr[X_i = 1] = p$
- $\Pr[X_i = 0] = 1 - p$
- Verteilung von $\sum_{i=1}^{n} X_i$:



Binomialverteilung

n Wiederholungen eines Bernoulli-Experiments:

- $\Pr[X_i = 1] = p$
- $\Pr[X_i = 0] = 1 - p$
- Verteilung von $\sum_{i=1}^{n} X_i$:



Chernoff Bound

Theorem (Chernoff Bound)

Seien $X_1, \ldots, X_n$ *unabhängige binäre Zufallsvariablen mit* $\Pr[X_i = 1] = p$ *und* $\Pr[X_i = 0] = 1 - p$ *(und somit* $\mu := \mathrm{Ex}[\sum_{i=1}^{n} X_i] = pn$*). Dann gilt*

Chernoff Bound

Theorem (Chernoff Bound)

Seien $X_1, \ldots, X_n$ unabhängige binäre Zufallsvariablen mit $\Pr[X_i = 1] = p$ und $\Pr[X_i = 0] = 1 - p$ (und somit $\mu := \mathrm{Ex}[\sum_{i=1}^{n} X_i] = pn$). Dann gilt

1. *Für jedes $\delta > 0$:*

$$\Pr\left[\sum_{i=1}^{n} X_i \geq (1 + \delta) \cdot \mu\right] \leq \frac{1}{e^{\frac{\min\{\delta, \delta^2\}}{3} \cdot \mu}}$$

Chernoff Bound

Theorem (Chernoff Bound)

Seien $X_1, \ldots, X_n$ *unabhängige binäre Zufallsvariablen mit* $\Pr[X_i = 1] = p$ *und* $\Pr[X_i = 0] = 1 - p$ *(und somit* $\mu := \mathrm{Ex}[\sum_{i=1}^n X_i] = pn$*). Dann gilt*

1. *Für jedes* $\delta > 0$*:*

$$\Pr\left[\sum_{i=1}^{n} X_i \geq (1+\delta)\cdot\mu\right] \leq \frac{1}{e^{\frac{\min\{\delta,\delta^2\}}{3}\cdot\mu}}$$

2. *Für jedes* $\delta \in [0,1]$*:*

$$\Pr\left[\sum_{i=1}^{n} X_i \leq (1-\delta)\cdot\mu\right] \leq \frac{1}{e^{\frac{\delta^2}{2}\cdot\mu}}$$

Maximal Independent Set

Definition

In einem ungerichteten Graph $G = (V, E)$ ist ein **Independent Set** eine Teilmenge von Knoten $U \subseteq V$, in der keine zwei Knoten benachbart sind.

Maximal Independent Set

Definition

In einem ungerichteten Graph $G = (V, E)$ ist ein **Independent Set** eine Teilmenge von Knoten $U \subseteq V$, in der keine zwei Knoten benachbart sind.

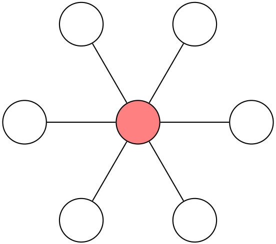
- Maxi**mal** Independent Set: Es kann kein Knoten zu U hinzugefügt werden, so dass U immer noch ein Independent Set ist
- Maxi**mum** Independent Set: U hat größtmögliche Kardinalität unter allen Independent Sets

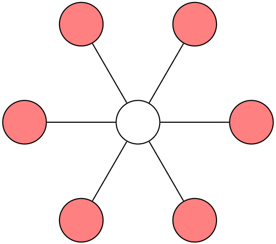
Maximal Independent Set

Definition

In einem ungerichteten Graph $G = (V, E)$ ist ein **Independent Set** eine Teilmenge von Knoten $U \subseteq V$, in der keine zwei Knoten benachbart sind.

- Maxi**mal** Independent Set: Es kann kein Knoten zu U hinzugefügt werden, so dass U immer noch ein Independent Set ist
- Maxi**mum** Independent Set: U hat größtmögliche Kardinalität unter allen Independent Sets





Sequentieller Algorithmus

Greedy:

$U \leftarrow \emptyset$
foreach $v \in V$ **do**
    if $\forall u \in U : (u, v) \notin E$ **then**
        $U \leftarrow U \cup \{v\}$
    end
end

Sequentieller Algorithmus

Greedy:

```
U ← ∅
foreach v ∈ V do
    if ∀u ∈ U : (u, v) ∉ E then
        U ← U ∪ {v}
    end
end
```

RAM Modell: Geeignete Implementierung benötigt Zeit $O(m + n)$

Sequentieller Algorithmus

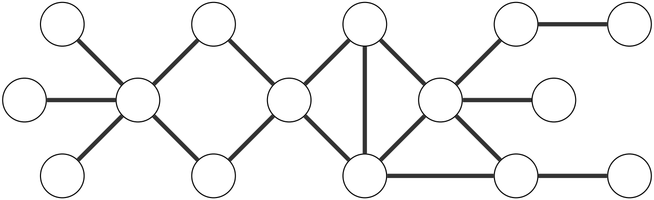
Greedy:

```
U ← ∅
foreach v ∈ V do
    if ∀u ∈ U : (u, v) ∉ E then
        U ← U ∪ {v}
    end
end
```

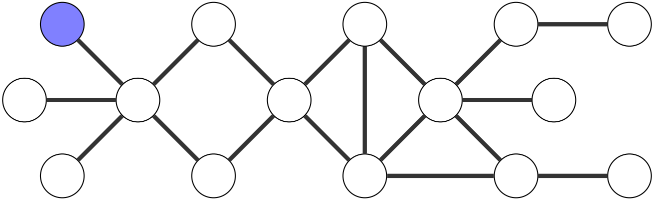
RAM Modell: Geeignete Implementierung benötigt Zeit $O(m + n)$

CONGEST Modell: Benötigt im schlechtesten Fall $\Omega(n)$ Runden

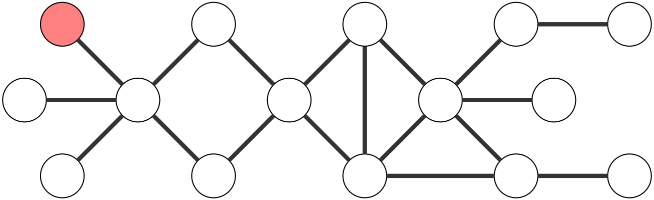
Beispiel



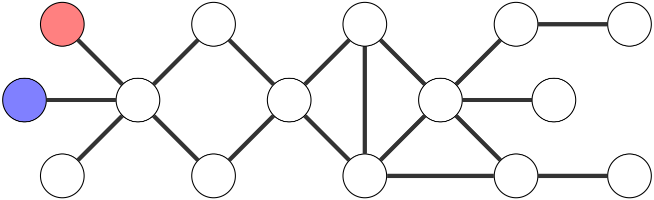
Beispiel



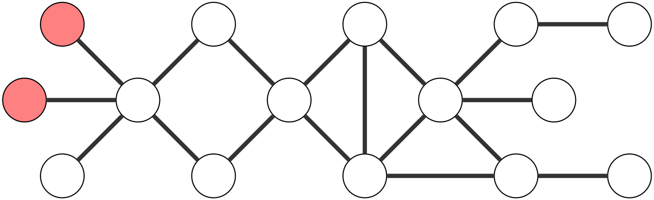
Beispiel



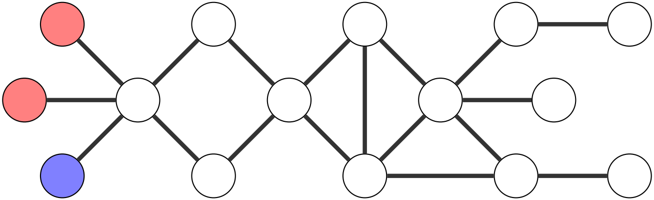
Beispiel



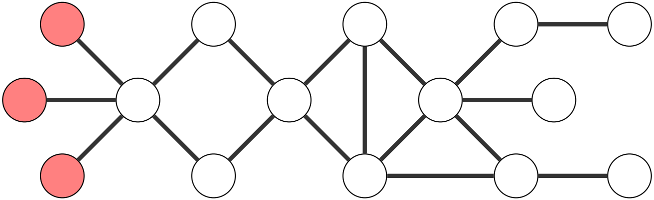
Beispiel



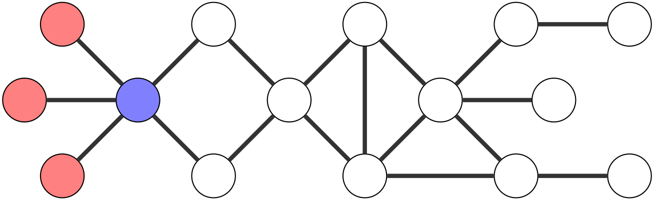
Beispiel



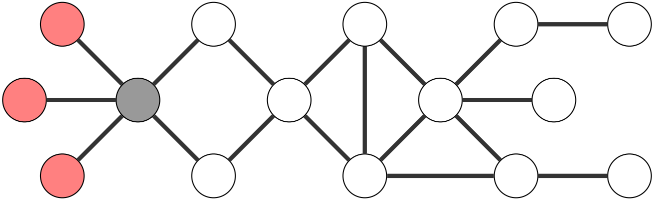
Beispiel



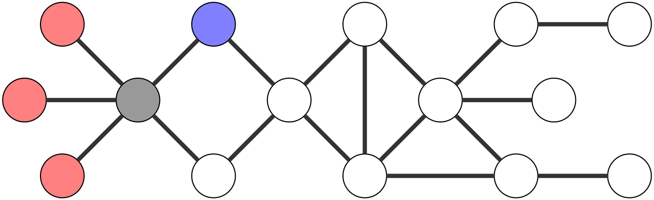
Beispiel



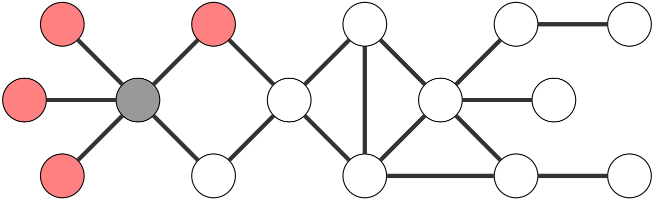
Beispiel



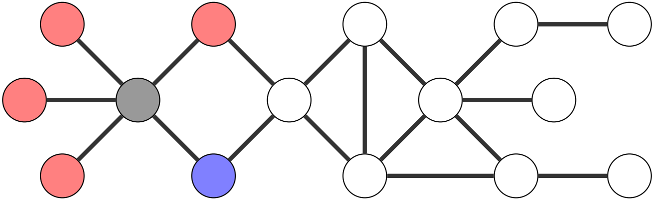
Beispiel



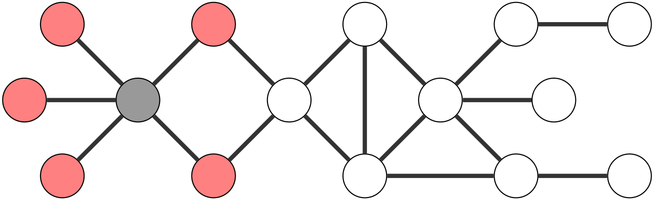
Beispiel



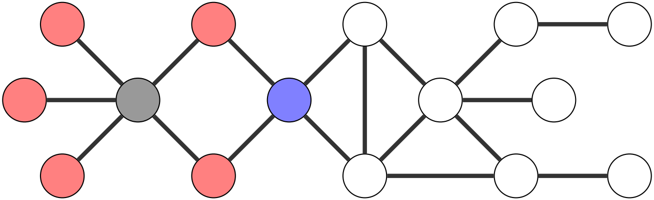
Beispiel



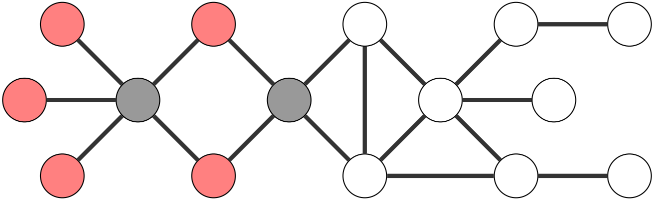
Beispiel



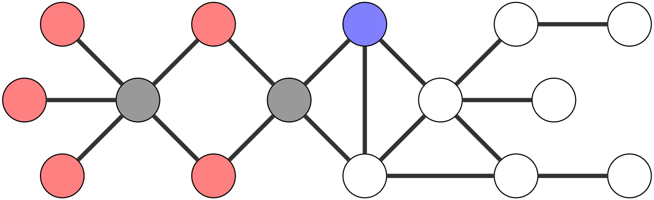
Beispiel



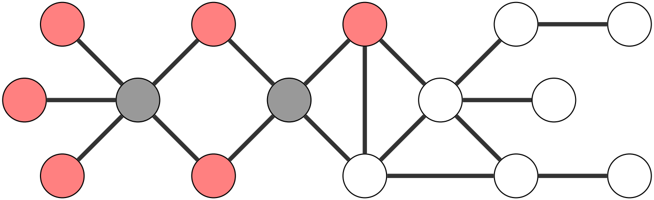
Beispiel



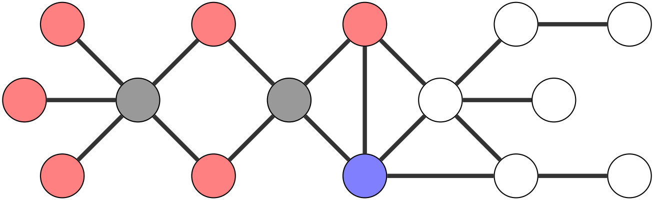
Beispiel



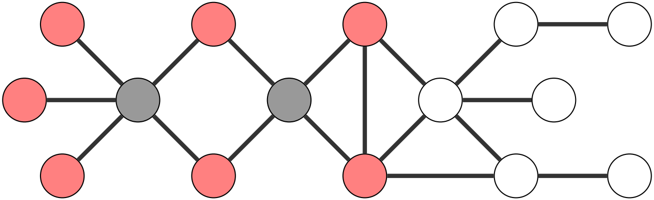
Beispiel



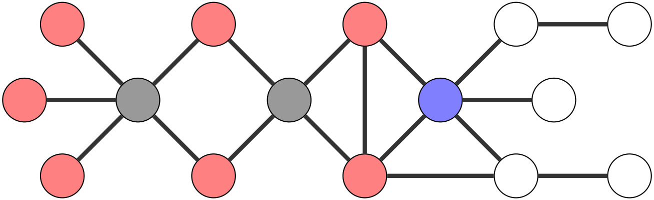
Beispiel



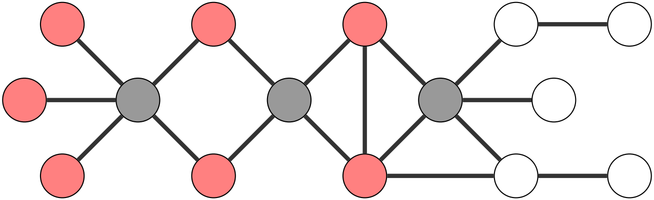
Beispiel



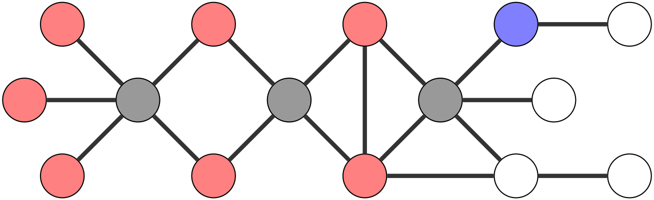
Beispiel



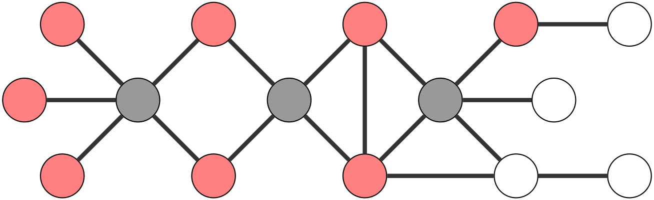
Beispiel



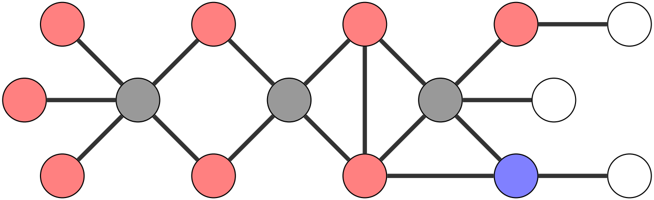
Beispiel



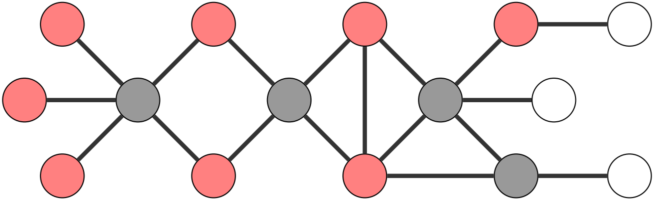
Beispiel



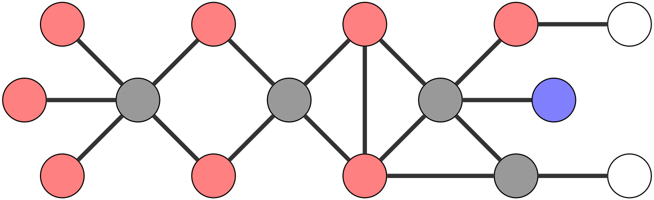
Beispiel



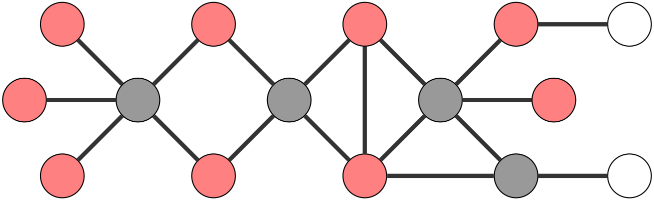
Beispiel



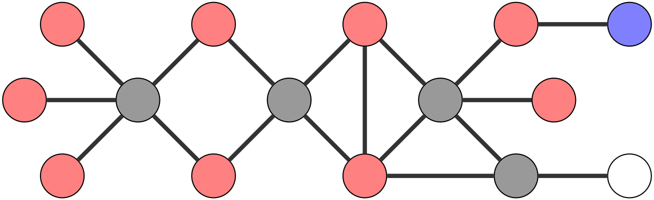
Beispiel



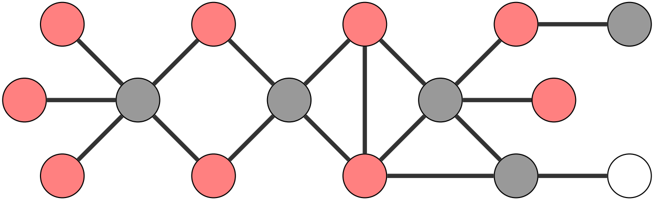
Beispiel



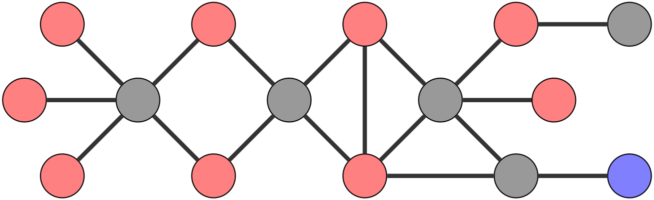
Beispiel



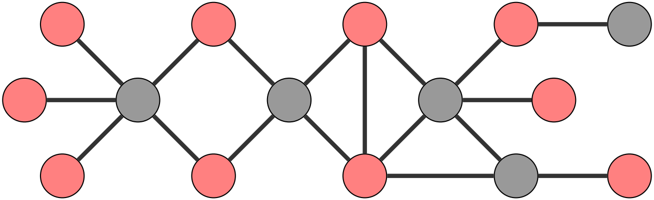
Beispiel



Beispiel



Beispiel



Verteilter Algorithmus [Luby ’85, Alon et al. ’86, Métivier et al. ’11]

Ziel: Jeder Knoten weiß, ob er in U ist oder nicht

Verteilter Algorithmus [Luby ’85, Alon et al. ’86, Métivier et al. ’11]

Ziel: Jeder Knoten weiß, ob er in U ist oder nicht

Jeder Knoten v führt folgenden Algorithmus aus:

while v *nicht terminiert* **do**
    Wähle uniforme reelle Zufallszahl $r(v) \in [0, 1($
    Sende $r(v)$ an alle Nachbarn
    Empfange $r(u)$ von jedem Nachbar u

Verteilter Algorithmus [Luby '85, Alon et al. '86, Métivier et al. '11]

Ziel: Jeder Knoten weiß, ob er in U ist oder nicht

Jeder Knoten v führt folgenden Algorithmus aus:

while v *nicht terminiert* **do**
    Wähle uniforme reelle Zufallszahl $r(v) \in [0, 1($
    Sende $r(v)$ an alle Nachbarn
    Empfange $r(u)$ von jedem Nachbar u
    if $r(v) < r(u)$ *für jeden Nachbar* u **then**
        Füge v zur Menge U hinzu
        Benachrichtige Nachbarn, dass $v \in U$
        Terminiere v
    end

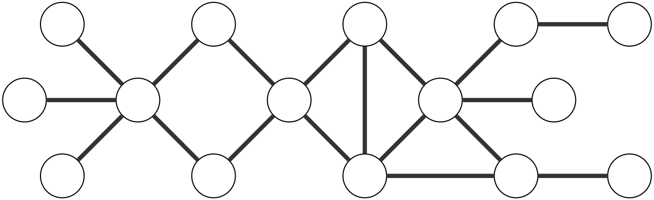
Verteilter Algorithmus [Luby ’85, Alon et al. ’86, Métivier et al. ’11]

Ziel: Jeder Knoten weiß, ob er in U ist oder nicht

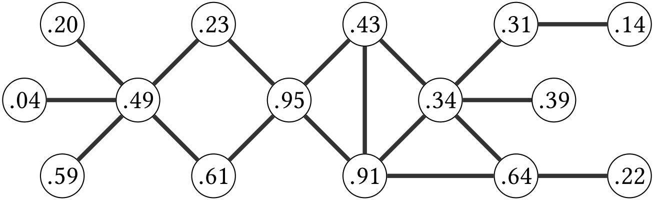
Jeder Knoten v führt folgenden Algorithmus aus:

while v *nicht terminiert* **do**
    Wähle uniforme reelle Zufallszahl $r(v) \in [0, 1($
    Sende $r(v)$ an alle Nachbarn
    Empfange $r(u)$ von jedem Nachbar u
    if $r(v) < r(u)$ *für jeden Nachbar* u **then**
        Füge v zur Menge U hinzu
        Benachrichtige Nachbarn, dass $v \in U$
        Terminiere v
    end
    Empfange Nachrichten über zu U hinzugefügte Nachbarn
    if *mindestens ein Nachbar wurde zu* U *hinzugefügt* **then**
        Terminiere v
    end
end

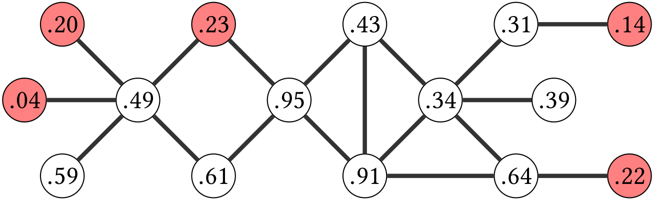
Beispiel



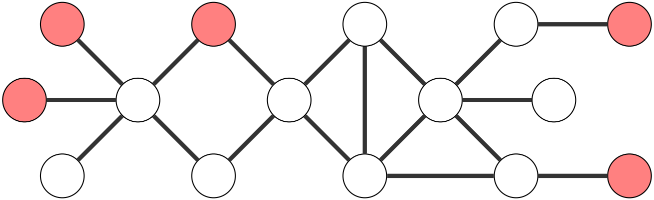
Beispiel



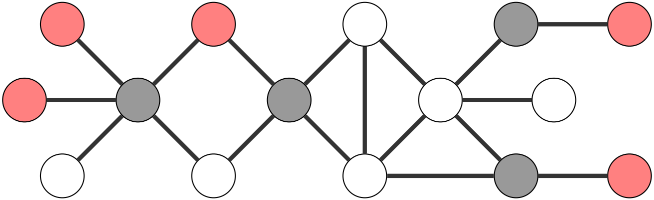
Beispiel



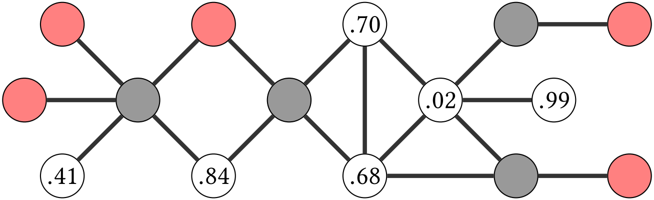
Beispiel



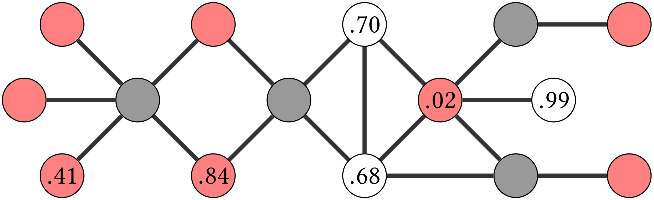
Beispiel



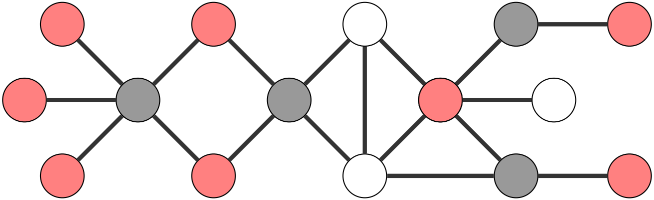
Beispiel



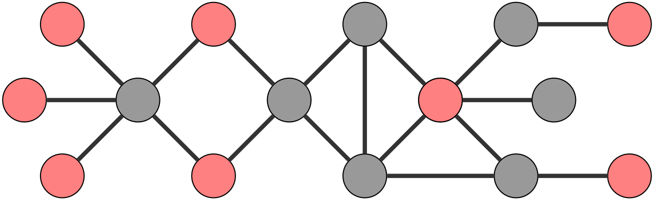
Beispiel



Beispiel



Beispiel



Korrektheit

Korrektheit

U ist Independent Set:

Korrektheit

U ist Independent Set:

- Induktion über Phasen (Iterationen der While-Schleife)

Korrektheit

U **ist Independent Set:**

- Induktion über Phasen (Iterationen der While-Schleife)
- Induktionshypothese: Knoten in U sind Independent Set und Menge der terminierten Knoten besteht aus U und Nachbarn von U

Korrektheit

U **ist Independent Set:**

- Induktion über Phasen (Iterationen der While-Schleife)
- Induktionshypothese: Knoten in U sind Independent Set und Menge der terminierten Knoten besteht aus U und Nachbarn von U
- Jeder Knoten, der zu U hinzugefügt wird, hat keine Nachbarn in U aus der aktuellen Phase und keine Nachbarn in U aus einer früheren Phase

Korrektheit

U **ist Independent Set:**

- Induktion über Phasen (Iterationen der While-Schleife)
- Induktionshypothese: Knoten in U sind Independent Set und Menge der terminierten Knoten besteht aus U und Nachbarn von U
- Jeder Knoten, der zu U hinzugefügt wird, hat keine Nachbarn in U aus der aktuellen Phase und keine Nachbarn in U aus einer früheren Phase

Maximalität:

Korrektheit

U ist Independent Set:

- Induktion über Phasen (Iterationen der While-Schleife)
- Induktionshypothese: Knoten in U sind Independent Set und Menge der terminierten Knoten besteht aus U und Nachbarn von U
- Jeder Knoten, der zu U hinzugefügt wird, hat keine Nachbarn in U aus der aktuellen Phase und keine Nachbarn in U aus einer früheren Phase

Maximalität:

- Knoten terminiert nur, wenn er selbst oder Nachbar in U ist

Korrektheit

U ist Independent Set:

- Induktion über Phasen (Iterationen der While-Schleife)
- Induktionshypothese: Knoten in U sind Independent Set und Menge der terminierten Knoten besteht aus U und Nachbarn von U
- Jeder Knoten, der zu U hinzugefügt wird, hat keine Nachbarn in U aus der aktuellen Phase und keine Nachbarn in U aus einer früheren Phase

Maximalität:

- Knoten terminiert nur, wenn er selbst oder Nachbar in U ist
- Sobald alle Knoten terminiert sind, kann kein Knoten mehr hinzugefügt werden, ohne Independent Set Eigenschaft zu verletzen

Korrektheit

U ist Independent Set:

- Induktion über Phasen (Iterationen der While-Schleife)
- Induktionshypothese: Knoten in U sind Independent Set und Menge der terminierten Knoten besteht aus U und Nachbarn von U
- Jeder Knoten, der zu U hinzugefügt wird, hat keine Nachbarn in U aus der aktuellen Phase und keine Nachbarn in U aus einer früheren Phase

Maximalität:

- Knoten terminiert nur, wenn er selbst oder Nachbar in U ist
- Sobald alle Knoten terminiert sind, kann kein Knoten mehr hinzugefügt werden, ohne Independent Set Eigenschaft zu verletzen

Terminierung: Mit Wahrscheinlichkeit 1

Korrektheit

U ist Independent Set:

- Induktion über Phasen (Iterationen der While-Schleife)
- Induktionshypothese: Knoten in U sind Independent Set und Menge der terminierten Knoten besteht aus U und Nachbarn von U
- Jeder Knoten, der zu U hinzugefügt wird, hat keine Nachbarn in U aus der aktuellen Phase und keine Nachbarn in U aus einer früheren Phase

Maximalität:

- Knoten terminiert nur, wenn er selbst oder Nachbar in U ist
- Sobald alle Knoten terminiert sind, kann kein Knoten mehr hinzugefügt werden, ohne Independent Set Eigenschaft zu verletzen

Terminierung: Mit Wahrscheinlichkeit 1

- Wahrscheinlichkeit, dass zwei zufällige reelle Zahlen gleich sind $= 0$

Korrektheit

U ist Independent Set:

- Induktion über Phasen (Iterationen der While-Schleife)
- Induktionshypothese: Knoten in U sind Independent Set und Menge der terminierten Knoten besteht aus U und Nachbarn von U
- Jeder Knoten, der zu U hinzugefügt wird, hat keine Nachbarn in U aus der aktuellen Phase und keine Nachbarn in U aus einer früheren Phase

Maximalität:

- Knoten terminiert nur, wenn er selbst oder Nachbar in U ist
- Sobald alle Knoten terminiert sind, kann kein Knoten mehr hinzugefügt werden, ohne Independent Set Eigenschaft zu verletzen

Terminierung: Mit Wahrscheinlichkeit 1

- Wahrscheinlichkeit, dass zwei zufällige reelle Zahlen gleich sind $= 0$
- Mit Union Bound: Wahrscheinlichkeit, dass Zufallszahlen in n Phasen unterschiedlich sind $= 1$

Korrektheit

U ist Independent Set:

- Induktion über Phasen (Iterationen der While-Schleife)
- Induktionshypothese: Knoten in U sind Independent Set und Menge der terminierten Knoten besteht aus U und Nachbarn von U
- Jeder Knoten, der zu U hinzugefügt wird, hat keine Nachbarn in U aus der aktuellen Phase und keine Nachbarn in U aus einer früheren Phase

Maximalität:

- Knoten terminiert nur, wenn er selbst oder Nachbar in U ist
- Sobald alle Knoten terminiert sind, kann kein Knoten mehr hinzugefügt werden, ohne Independent Set Eigenschaft zu verletzen

Terminierung: Mit Wahrscheinlichkeit 1

- Wahrscheinlichkeit, dass zwei zufällige reelle Zahlen gleich sind $= 0$
- Mit Union Bound: Wahrscheinlichkeit, dass Zufallszahlen in n Phasen unterschiedlich sind $= 1$
- In jeder Phase wird zumindest Knoten mit kleinstem Wert für $r(v)$ der Menge U hinzugefügt

Laufzeitanalyse

Strategie:

- Analysiere Reduktion der Anzahl an Kanten im von nicht-terminierten Knoten induzierten Subgraph

Laufzeitanalyse

Strategie:

- Analysiere Reduktion der Anzahl an Kanten im von nicht-terminierten Knoten induzierten Subgraph
- Zeige: In jeder Phase wird die Anzahl der Kanten in Erwartung um (mindestens) die Hälfte reduziert

Laufzeitanalyse

Strategie:

- Analysiere Reduktion der Anzahl an Kanten im von nicht-terminierten Knoten induzierten Subgraph
- Zeige: In jeder Phase wird die Anzahl der Kanten in Erwartung um (mindestens) die Hälfte reduziert
- Anwendung von Chernoff und Markov Bound: $O(\log n)$ Phasen
  Jede Phase dauert $O(1)$ Runden $\Rightarrow O(\log n)$ Runden

Laufzeitanalyse

Strategie:

- Analysiere Reduktion der Anzahl an Kanten im von nicht-terminierten Knoten induzierten Subgraph
- Zeige: In jeder Phase wird die Anzahl der Kanten in Erwartung um (mindestens) die Hälfte reduziert
- Anwendung von Chernoff und Markov Bound: $O(\log n)$ Phasen
  Jede Phase dauert $O(1)$ Runden $\Rightarrow O(\log n)$ Runden
- Implementierungsdetail: Übertragung von $O(\log n)$ Bits der reellen Zufallszahlen

Erwartungswert

Analysetrick:

- Betrachte jede Kante als zwei gerichtete Kanten
- Systematische Unterschätzung entfernter Kanten erleichtert Analyse

Erwartungswert

Analysetrick:

- Betrachte jede Kante als zwei gerichtete Kanten
- Systematische Unterschätzung entfernter Kanten erleichtert Analyse

u **dominiert** v ($u \twoheadrightarrow v$) falls

1. $r(u) < r(u')$ für alle Nachbarn u' von u und

Erwartungswert

Analysetrick:

- Betrachte jede Kante als zwei gerichtete Kanten
- Systematische Unterschätzung entfernter Kanten erleichtert Analyse

u **dominiert** v ($u \twoheadrightarrow v$) falls

1. $r(u) < r(u')$ für alle Nachbarn u' von u und
2. $r(u) < r(v')$ für alle Nachbarn $v' \neq u$ von v

Erwartungswert

Analysetrick:

- Betrachte jede Kante als zwei gerichtete Kanten
- Systematische Unterschätzung entfernter Kanten erleichtert Analyse

u **dominiert** v ($u \twoheadrightarrow v$) falls

1. $r(u) < r(u')$ für alle Nachbarn u' von u und
2. $r(u) < r(v')$ für alle Nachbarn $v' \neq u$ von v

Idee: Wenn $u \twoheadrightarrow v$, dann terminiert v und ausgehende Kanten von v werden aus Subgraph nicht-terminierter Knoten entfernt

Erwartungswert

Analysetrick:

- Betrachte jede Kante als zwei gerichtete Kanten
- Systematische Unterschätzung entfernter Kanten erleichtert Analyse

u **dominiert** v ($u \twoheadrightarrow v$) falls

1. $r(u) < r(u')$ für alle Nachbarn u' von u und
2. $r(u) < r(v')$ für alle Nachbarn $v' \neq u$ von v

Idee: Wenn $u \twoheadrightarrow v$, dann terminiert v und ausgehende Kanten von v werden aus Subgraph nicht-terminierter Knoten entfernt

$$\Pr[u \twoheadrightarrow v] \geq \frac{1}{\deg(u) + \deg(v)}$$

Erwartungswert

Analysetrick:

- Betrachte jede Kante als zwei gerichtete Kanten
- Systematische Unterschätzung entfernter Kanten erleichtert Analyse

u **dominiert** v ($u \twoheadrightarrow v$) falls

1. $r(u) < r(u')$ für alle Nachbarn u' von u und
2. $r(u) < r(v')$ für alle Nachbarn $v' \neq u$ von v

Idee: Wenn $u \twoheadrightarrow v$, dann terminiert v und ausgehende Kanten von v werden aus Subgraph nicht-terminierter Knoten entfernt

$$\Pr[u \twoheadrightarrow v] \geq \frac{1}{\deg(u) + \deg(v)}$$

Begründung:

- Anzahl Knoten in Nachbarschaften von u und v: $\leq \deg(u) + \deg(v)$

Erwartungswert

Analysetrick:

- Betrachte jede Kante als zwei gerichtete Kanten
- Systematische Unterschätzung entfernter Kanten erleichtert Analyse

u **dominiert** v ($u \twoheadrightarrow v$) falls

1. $r(u) < r(u')$ für alle Nachbarn u' von u und
2. $r(u) < r(v')$ für alle Nachbarn $v' \neq u$ von v

Idee: Wenn $u \twoheadrightarrow v$, dann terminiert v und ausgehende Kanten von v werden aus Subgraph nicht-terminierter Knoten entfernt

$$\Pr[u \twoheadrightarrow v] \geq \frac{1}{\deg(u) + \deg(v)}$$

Begründung:

- Anzahl Knoten in Nachbarschaften von u und v: $\leq \deg(u) + \deg(v)$
- Zufallszahlen $r(\cdot)$ induzieren uniform zufällige Permutation von V

Erwartungswert

Analysetrick:

- Betrachte jede Kante als zwei gerichtete Kanten
- Systematische Unterschätzung entfernter Kanten erleichtert Analyse

u **dominiert** v ($u \twoheadrightarrow v$) falls

1. $r(u) < r(u')$ für alle Nachbarn u' von u und
2. $r(u) < r(v')$ für alle Nachbarn $v' \neq u$ von v

Idee: Wenn $u \twoheadrightarrow v$, dann terminiert v und ausgehende Kanten von v werden aus Subgraph nicht-terminierter Knoten entfernt

$$\Pr[u \twoheadrightarrow v] \geq \frac{1}{\deg(u) + \deg(v)}$$

Begründung:

- Anzahl Knoten in Nachbarschaften von u und v: $\leq \deg(u) + \deg(v)$
- Zufallszahlen $r(\cdot)$ induzieren uniform zufällige Permutation von V
- Wahrscheinlichkeit dass u erster in Permutation ist: $1/\#$Knoten

Erwartungswert (Fortsetzung)

F: Menge ungerichteter Kanten am Beginn der aktuellen Phase

X: Zufallsvariable für Anzahl entfernter *gerichteter* Kanten

Erwartungswert (Fortsetzung)

F: Menge ungerichteter Kanten am Beginn der aktuellen Phase
X: Zufallsvariable für Anzahl entfernter *gerichteter* Kanten

$X_{(u \twoheadrightarrow v)}$ = # ausgehender Kanten von v, die wegen $u \twoheadrightarrow v$ entfernt werden

Erwartungswert (Fortsetzung)

F: Menge ungerichteter Kanten am Beginn der aktuellen Phase
X: Zufallsvariable für Anzahl entfernter *gerichteter* Kanten

$X_{(u \twoheadrightarrow v)}$ = # ausgehender Kanten von v, die wegen $u \twoheadrightarrow v$ entfernt werden

$$\mathrm{Ex}[X] \geq \sum_{\{u,v\} \in F} \left(\mathrm{Ex}[X_{(u \twoheadrightarrow v)}] + \mathrm{Ex}[X_{(v \twoheadrightarrow u)}]\right)$$

Erwartungswert (Fortsetzung)

F: Menge ungerichteter Kanten am Beginn der aktuellen Phase

X: Zufallsvariable für Anzahl entfernter *gerichteter* Kanten

$X_{(u \twoheadrightarrow v)}$ = # ausgehender Kanten von v, die wegen $u \twoheadrightarrow v$ entfernt werden

$$\begin{aligned}
\mathrm{Ex}[X] &\geq \sum_{\{u,v\} \in F} \left(\mathrm{Ex}[X_{(u \twoheadrightarrow v)}] + \mathrm{Ex}[X_{(v \twoheadrightarrow u)}]\right) \\
&= \sum_{\{u,v\} \in F} \left(\Pr[X_{(u \twoheadrightarrow v)}] \cdot \deg(v) + \Pr[X_{(v \twoheadrightarrow u)}] \cdot \deg(u)\right)
\end{aligned}$$

Erwartungswert (Fortsetzung)

F: Menge ungerichteter Kanten am Beginn der aktuellen Phase

X: Zufallsvariable für Anzahl entfernter *gerichteter* Kanten

$X_{(u \twoheadrightarrow v)}$ = # ausgehender Kanten von v, die wegen $u \twoheadrightarrow v$ entfernt werden

$$\begin{aligned}
\operatorname{Ex}[X] &\geq \sum_{\{u,v\} \in F} \left(\operatorname{Ex}[X_{(u \twoheadrightarrow v)}] + \operatorname{Ex}[X_{(v \twoheadrightarrow u)}]\right) \\
&= \sum_{\{u,v\} \in F} \left(\Pr[X_{(u \twoheadrightarrow v)}] \cdot \deg(v) + \Pr[X_{(v \twoheadrightarrow u)}] \cdot \deg(u)\right) \\
&\geq \sum_{\{u,v\} \in E} \left(\frac{\deg(v)}{\deg(u) + \deg(v)} + \frac{\deg(u)}{\deg(v) + \deg(u)}\right)
\end{aligned}$$

Erwartungswert (Fortsetzung)

F: Menge ungerichteter Kanten am Beginn der aktuellen Phase

X: Zufallsvariable für Anzahl entfernter *gerichteter* Kanten

$X_{(u \twoheadrightarrow v)}$ = # ausgehender Kanten von v, die wegen $u \twoheadrightarrow v$ entfernt werden

$$\begin{aligned}
\mathrm{Ex}[X] &\geq \sum_{\{u,v\} \in F} \left(\mathrm{Ex}[X_{(u \twoheadrightarrow v)}] + \mathrm{Ex}[X_{(v \twoheadrightarrow u)}]\right) \\
&= \sum_{\{u,v\} \in F} \left(\Pr[X_{(u \twoheadrightarrow v)}] \cdot \deg(v) + \Pr[X_{(v \twoheadrightarrow u)}] \cdot \deg(u)\right) \\
&\geq \sum_{\{u,v\} \in E} \left(\frac{\deg(v)}{\deg(u) + \deg(v)} + \frac{\deg(u)}{\deg(v) + \deg(u)}\right) \\
&= \sum_{\{u,v\} \in F} 1 = |F|
\end{aligned}$$

Erwartungswert (Fortsetzung)

F: Menge ungerichteter Kanten am Beginn der aktuellen Phase
X: Zufallsvariable für Anzahl entfernter *gerichteter* Kanten

$X_{(u \twoheadrightarrow v)}$ = # ausgehender Kanten von v, die wegen $u \twoheadrightarrow v$ entfernt werden

$$\begin{aligned}
\mathrm{Ex}[X] &\geq \sum_{\{u,v\} \in F} \left(\mathrm{Ex}[X_{(u \twoheadrightarrow v)}] + \mathrm{Ex}[X_{(v \twoheadrightarrow u)}]\right) \\
&= \sum_{\{u,v\} \in F} \left(\Pr[X_{(u \twoheadrightarrow v)}] \cdot \deg(v) + \Pr[X_{(v \twoheadrightarrow u)}] \cdot \deg(u)\right) \\
&\geq \sum_{\{u,v\} \in E} \left(\frac{\deg(v)}{\deg(u) + \deg(v)} + \frac{\deg(u)}{\deg(v) + \deg(u)}\right) \\
&= \sum_{\{u,v\} \in F} 1 = |F|
\end{aligned}$$

Es werden mindestens halb so viele *ungerichtete* Kanten entfernt!

Erwartungswert (Fortsetzung)

F: Menge ungerichteter Kanten am Beginn der aktuellen Phase
X: Zufallsvariable für Anzahl entfernter *gerichteter* Kanten

$X_{(u \twoheadrightarrow v)}$ = # ausgehender Kanten von v, die wegen $u \twoheadrightarrow v$ entfernt werden

$$\begin{aligned}
\mathrm{Ex}[X] &\geq \sum_{\{u,v\} \in F} \left(\mathrm{Ex}[X_{(u \twoheadrightarrow v)}] + \mathrm{Ex}[X_{(v \twoheadrightarrow u)}] \right) \\
&= \sum_{\{u,v\} \in F} \left(\Pr[X_{(u \twoheadrightarrow v)}] \cdot \deg(v) + \Pr[X_{(v \twoheadrightarrow u)}] \cdot \deg(u) \right) \\
&\geq \sum_{\{u,v\} \in E} \left(\frac{\deg(v)}{\deg(u) + \deg(v)} + \frac{\deg(u)}{\deg(v) + \deg(u)} \right) \\
&= \sum_{\{u,v\} \in F} 1 = |F|
\end{aligned}$$

Es werden mindestens halb so viele *ungerichtete* Kanten entfernt!

Lemma

In jeder Phase wird in Erwartung mindestens die Hälfte der Kanten entfernt.

Markov Bound

Theorem (Markov Bound)

Sei X eine nicht-negative Zufallsvariable und $\alpha > 0$. Dann gilt

$$\Pr[Y \geq \alpha \cdot \mathrm{Ex}[Y]] \leq \frac{1}{\alpha} .$$

Markov Bound

Theorem (Markov Bound)

Sei X eine nicht-negative Zufallsvariable und $\alpha > 0$. Dann gilt

$$\Pr[Y \geq \alpha \cdot \mathrm{Ex}[Y]] \leq \frac{1}{\alpha} .$$

Y: Zufallsvariable für Anzahl der nicht-entfernten Kanten

Markov Bound

Theorem (Markov Bound)

Sei X eine nicht-negative Zufallsvariable und $\alpha > 0$. Dann gilt

$$\Pr[Y \geq \alpha \cdot \mathrm{Ex}[Y]] \leq \frac{1}{\alpha} .$$

Y: Zufallsvariable für Anzahl der nicht-entfernten Kanten

Wir wissen: $\mathrm{Ex}[Y] \leq \frac{|F|}{2}$

Markov Bound

Theorem (Markov Bound)

Sei X eine nicht-negative Zufallsvariable und $\alpha > 0$. Dann gilt

$$\Pr[Y \geq \alpha \cdot \mathrm{Ex}[Y]] \leq \frac{1}{\alpha}\,.$$

Y: Zufallsvariable für Anzahl der nicht-entfernten Kanten

Wir wissen: $\mathrm{Ex}[Y] \leq \frac{|F|}{2}$
Somit:

$$\Pr\left[Y \geq \frac{2}{3} \cdot |F|\right] \leq \Pr\left[Y \geq \frac{4}{3} \cdot \mathrm{Ex}[Y]\right] \leq \frac{3}{4}$$

Markov Bound

Theorem (Markov Bound)

Sei X eine nicht-negative Zufallsvariable und $\alpha > 0$. Dann gilt

$$\Pr[Y \geq \alpha \cdot \operatorname{Ex}[Y]] \leq \frac{1}{\alpha}\,.$$

Y: Zufallsvariable für Anzahl der nicht-entfernten Kanten

Wir wissen: $\operatorname{Ex}[Y] \leq \frac{|F|}{2}$

Somit:

$$\Pr\left[Y \geq \frac{2}{3} \cdot |F|\right] \leq \Pr\left[Y \geq \frac{4}{3} \cdot \operatorname{Ex}[Y]\right] \leq \frac{3}{4}$$

Gegenereignis:

$$\Pr\left[Y < \frac{2}{3} \cdot |F|\right] = 1 - \Pr\left[Y \geq \frac{2}{3} \cdot |F|\right] \geq 1 - \frac{3}{4} = \frac{1}{4}$$

Markov Bound

Theorem (Markov Bound)

Sei X eine nicht-negative Zufallsvariable und $\alpha > 0$. Dann gilt

$$\Pr[Y \geq \alpha \cdot \mathrm{Ex}[Y]] \leq \frac{1}{\alpha}\,.$$

Y: Zufallsvariable für Anzahl der nicht-entfernten Kanten

Wir wissen: $\mathrm{Ex}[Y] \leq \frac{|F|}{2}$

Somit:

$$\Pr\left[Y \geq \frac{2}{3} \cdot |F|\right] \leq \Pr\left[Y \geq \frac{4}{3} \cdot \mathrm{Ex}[Y]\right] \leq \frac{3}{4}$$

Gegenereignis:

$$\Pr\left[Y < \frac{2}{3} \cdot |F|\right] = 1 - \Pr\left[Y \geq \frac{2}{3} \cdot |F|\right] \geq 1 - \frac{3}{4} = \frac{1}{4}$$

Lemma

In jeder Phase wird mit Wahrscheinlichkeit $\geq \frac{1}{4}$ mindestens ein Drittel der Kanten entfernt.

Chernoff Bound

Chernoff Bound

Seien $Z_1, \ldots, Z_\ell$ *unabhängige* binäre Zufallsvariablen mit $\Pr[Z_i = 1] = p$ und $\Pr[Z_i = 0] = 1 - p$ (und somit $\mu := \mathrm{Ex}[\sum_{i=1}^{\ell} Z_i] = p\ell$). Dann gilt für jedes $\delta \in [0, 1]$:

$$\Pr\left[\sum_{i=1}^{\ell} Z_i \leq (1 - \delta) \cdot \mu\right] \leq \frac{1}{e^{\frac{\delta^2}{2}\mu}}$$

Anzahl Phasen

Lemma

Mit hoher Wahrscheinlichkeit terminiert Algorithmus nach $32\lceil c \ln n \rceil$ *Phasen.*

Anzahl Phasen

Lemma

Mit hoher Wahrscheinlichkeit terminiert Algorithmus nach $32\lceil c \ln n \rceil$ *Phasen.*

Beweis:

Idee: Unterteile in gute und schlechte Phasen, binäre Zufallsvariable

$$Z_i = \begin{cases} 1 & \text{falls in Phase } i \text{ mindestens } 1/3 \text{ der Kanten entfernt werden} \\ 0 & \text{andernfalls} \end{cases}$$

Anzahl Phasen

Lemma

Mit hoher Wahrscheinlichkeit terminiert Algorithmus nach $32\lceil c \ln n \rceil$ Phasen.

Beweis:

Idee: Unterteile in gute und schlechte Phasen, binäre Zufallsvariable

$$Z_i = \begin{cases} 1 & \text{falls in Phase } i \text{ mindestens } 1/3 \text{ der Kanten entfernt werden} \\ 0 & \text{andernfalls} \end{cases}$$

$Z = \sum_{i=1}^{32\lceil c \ln n \rceil} Z_i$ ist Zufallsvariable für Anzahl guter Phasen (mit $Z_i = 1$)
$\mathrm{Ex}[Z] = \frac{1}{4} \cdot 32\lceil c \ln n \rceil \geq 8 \ln n$

Anzahl Phasen

Lemma

Mit hoher Wahrscheinlichkeit terminiert Algorithmus nach $32\lceil c \ln n \rceil$ Phasen.

Beweis:

Idee: Unterteile in gute und schlechte Phasen, binäre Zufallsvariable

$$Z_i = \begin{cases} 1 & \text{falls in Phase } i \text{ mindestens } 1/3 \text{ der Kanten entfernt werden} \\ 0 & \text{andernfalls} \end{cases}$$

$Z = \sum_{i=1}^{32\lceil c \ln n \rceil} Z_i$ ist Zufallsvariable für Anzahl guter Phasen (mit $Z_i = 1$)
$\mathrm{Ex}[Z] = \frac{1}{4} \cdot 32\lceil c \ln n \rceil \geq 8 \ln n$

Nach k guten Phasen: #Kanten $\leq \frac{m}{3^k} \leq \frac{n^2}{3^k}$

Anzahl Phasen

Lemma

Mit hoher Wahrscheinlichkeit terminiert Algorithmus nach $32\lceil c \ln n \rceil$ Phasen.

Beweis:

Idee: Unterteile in gute und schlechte Phasen, binäre Zufallsvariable

$$Z_i = \begin{cases} 1 & \text{falls in Phase } i \text{ mindestens } 1/3 \text{ der Kanten entfernt werden} \\ 0 & \text{andernfalls} \end{cases}$$

$Z = \sum_{i=1}^{32\lceil c \ln n \rceil} Z_i$ ist Zufallsvariable für Anzahl guter Phasen (mit $Z_i = 1$)
$\mathrm{Ex}[Z] = \frac{1}{4} \cdot 32\lceil c \ln n \rceil \geq 8 \ln n$

Nach k guten Phasen: #Kanten $\leq \frac{m}{3^k} \leq \frac{n^2}{3^k}$
$\Rightarrow$ Für #Kanten < 1 sind $4 \ln n > \frac{2 \ln n}{\ln 3} + 1 = \log_3(n^2) + 1$ gute Phasen ausreichend;

Anzahl Phasen

Lemma

Mit hoher Wahrscheinlichkeit terminiert Algorithmus nach $32\lceil c \ln n \rceil$ *Phasen.*

Beweis:

Idee: Unterteile in gute und schlechte Phasen, binäre Zufallsvariable

$$Z_i = \begin{cases} 1 & \text{falls in Phase } i \text{ mindestens } 1/3 \text{ der Kanten entfernt werden} \\ 0 & \text{andernfalls} \end{cases}$$

$Z = \sum_{i=1}^{32\lceil c \ln n \rceil} Z_i$ ist Zufallsvariable für Anzahl guter Phasen (mit $Z_i = 1$)
$\mathrm{Ex}[Z] = \frac{1}{4} \cdot 32\lceil c \ln n \rceil \geq 8 \ln n$

Nach k guten Phasen: #Kanten $\leq \frac{m}{3^k} \leq \frac{n^2}{3^k}$
$\Rightarrow$ Für #Kanten < 1 sind $4 \ln n > \frac{2 \ln n}{\ln 3} + 1 = \log_3(n^2) + 1$ gute Phasen ausreichend; Chernoff-Bound für Gegenwahrscheinlichkeit (mit $\delta = \frac{1}{2}$):

Anzahl Phasen

Lemma

Mit hoher Wahrscheinlichkeit terminiert Algorithmus nach $32\lceil c \ln n \rceil$ *Phasen.*

Beweis:

Idee: Unterteile in gute und schlechte Phasen, binäre Zufallsvariable

$$Z_i = \begin{cases} 1 & \text{falls in Phase } i \text{ mindestens } 1/3 \text{ der Kanten entfernt werden} \\ 0 & \text{andernfalls} \end{cases}$$

$Z = \sum_{i=1}^{32\lceil c \ln n \rceil} Z_i$ ist Zufallsvariable für Anzahl guter Phasen (mit $Z_i = 1$)
$\mathrm{Ex}[Z] = \frac{1}{4} \cdot 32\lceil c \ln n \rceil \geq 8 \ln n$

Nach k guten Phasen: #Kanten $\leq \frac{m}{3^k} \leq \frac{n^2}{3^k}$
$\Rightarrow$ Für #Kanten < 1 sind $4 \ln n > \frac{2 \ln n}{\ln 3} + 1 = \log_3(n^2) + 1$ gute Phasen ausreichend; Chernoff-Bound für Gegenwahrscheinlichkeit (mit $\delta = \frac{1}{2}$):

$$\Pr[Z < 4 \ln n] \leq \Pr\left[Z < \frac{1}{2}\,\mathrm{Ex}[Z]\right]$$

Anzahl Phasen

Lemma

Mit hoher Wahrscheinlichkeit terminiert Algorithmus nach $32\lceil c \ln n \rceil$ *Phasen.*

Beweis:

Idee: Unterteile in gute und schlechte Phasen, binäre Zufallsvariable

$$Z_i = \begin{cases} 1 & \text{falls in Phase } i \text{ mindestens } 1/3 \text{ der Kanten entfernt werden} \\ 0 & \text{andernfalls} \end{cases}$$

$Z = \sum_{i=1}^{32\lceil c \ln n \rceil} Z_i$ ist Zufallsvariable für Anzahl guter Phasen (mit $Z_i = 1$)
$\mathrm{Ex}[Z] = \frac{1}{4} \cdot 32\lceil c \ln n \rceil \geq 8 \ln n$

Nach k guten Phasen: #Kanten $\leq \frac{m}{3^k} \leq \frac{n^2}{3^k}$
$\Rightarrow$ Für #Kanten < 1 sind $4 \ln n > \frac{2 \ln n}{\ln 3} + 1 = \log_3(n^2) + 1$ gute Phasen ausreichend; Chernoff-Bound für Gegenwahrscheinlichkeit (mit $\delta = \frac{1}{2}$):

$$\Pr[Z < 4 \ln n] \leq \Pr\left[Z < \frac{1}{2}\,\mathrm{Ex}[Z]\right] \leq \frac{1}{e^{\mathrm{Ex}[Z]/8}}$$

Anzahl Phasen

Lemma

Mit hoher Wahrscheinlichkeit terminiert Algorithmus nach $32\lceil c \ln n \rceil$ Phasen.

Beweis:

Idee: Unterteile in gute und schlechte Phasen, binäre Zufallsvariable

$$Z_i = \begin{cases} 1 & \text{falls in Phase } i \text{ mindestens } 1/3 \text{ der Kanten entfernt werden} \\ 0 & \text{andernfalls} \end{cases}$$

$Z = \sum_{i=1}^{32\lceil c \ln n \rceil} Z_i$ ist Zufallsvariable für Anzahl guter Phasen (mit $Z_i = 1$)
$\mathrm{Ex}[Z] = \frac{1}{4} \cdot 32\lceil c \ln n \rceil \geq 8 \ln n$

Nach k guten Phasen: #Kanten $\leq \frac{m}{3^k} \leq \frac{n^2}{3^k}$
$\Rightarrow$ Für #Kanten < 1 sind $4 \ln n > \frac{2 \ln n}{\ln 3} + 1 = \log_3(n^2) + 1$ gute Phasen ausreichend; Chernoff-Bound für Gegenwahrscheinlichkeit (mit $\delta = \frac{1}{2}$):

$$\Pr[Z < 4 \ln n] \leq \Pr\left[Z < \frac{1}{2} \mathrm{Ex}[Z]\right] \leq \frac{1}{e^{\mathrm{Ex}[Z]/8}} \leq \frac{1}{e^{c \ln n}}$$

Anzahl Phasen

Lemma

Mit hoher Wahrscheinlichkeit terminiert Algorithmus nach $32\lceil c \ln n \rceil$ *Phasen.*

Beweis:

Idee: Unterteile in gute und schlechte Phasen, binäre Zufallsvariable

$$Z_i = \begin{cases} 1 & \text{falls in Phase } i \text{ mindestens } 1/3 \text{ der Kanten entfernt werden} \\ 0 & \text{andernfalls} \end{cases}$$

$Z = \sum_{i=1}^{32\lceil c \ln n \rceil} Z_i$ ist Zufallsvariable für Anzahl guter Phasen (mit $Z_i = 1$)
$\text{Ex}[Z] = \frac{1}{4} \cdot 32\lceil c \ln n \rceil \geq 8 \ln n$

Nach k guten Phasen: #Kanten $\leq \frac{m}{3^k} \leq \frac{n^2}{3^k}$
$\Rightarrow$ Für #Kanten < 1 sind $4 \ln n > \frac{2 \ln n}{\ln 3} + 1 = \log_3(n^2) + 1$ gute Phasen ausreichend; Chernoff-Bound für Gegenwahrscheinlichkeit (mit $\delta = \frac{1}{2}$):

$$\Pr[Z < 4 \ln n] \leq \Pr\left[Z < \frac{1}{2}\text{Ex}[Z]\right] \leq \frac{1}{e^{\text{Ex}[Z]/8}} \leq \frac{1}{e^{c \ln n}} = \frac{1}{n^c}$$

Bandbreitenbeschränkung

Ziel: Für jede Kante $\{u, v\}$, entscheide, ob $r(u) < r(v)$ oder $r(u) > r(v)$

Bandbreitenbeschränkung

Ziel: Für jede Kante $\{u, v\}$, entscheide, ob $r(u) < r(v)$ oder $r(u) > r(v)$

Zufällige reelle Zahl $\in [0, 1($: unendlicher String zufälliger Bits

Bandbreitenbeschränkung

Ziel: Für jede Kante $\{u, v\}$, entscheide, ob $r(u) < r(v)$ oder $r(u) > r(v)$

Zufällige reelle Zahl $\in [0, 1($: unendlicher String zufälliger Bits
Idee: u und v tauschen nur die ersten $(c + 3)\lceil \log n \rceil$ Bits aus

Bandbreitenbeschränkung

Ziel: Für jede Kante $\{u, v\}$, entscheide, ob $r(u) < r(v)$ oder $r(u) > r(v)$

Zufällige reelle Zahl $\in [0, 1($: unendlicher String zufälliger Bits
Idee: u und v tauschen nur die ersten $(c + 3)\lceil \log n \rceil$ Bits aus

Wahrscheinlichkeit, dass die ersten $(c + 3)\lceil \log n \rceil$ Bits gleich sind:

Bandbreitenbeschränkung

Ziel: Für jede Kante $\{u, v\}$, entscheide, ob $r(u) < r(v)$ oder $r(u) > r(v)$

Zufällige reelle Zahl $\in [0, 1($: unendlicher String zufälliger Bits
Idee: u und v tauschen nur die ersten $(c + 3)\lceil \log n \rceil$ Bits aus

Wahrscheinlichkeit, dass die ersten $(c + 3)\lceil \log n \rceil$ Bits gleich sind:

$$\left(\frac{1}{2}\right)^{(c+3)\lceil \log n \rceil}$$

Bandbreitenbeschränkung

Ziel: Für jede Kante $\{u, v\}$, entscheide, ob $r(u) < r(v)$ oder $r(u) > r(v)$

Zufällige reelle Zahl $\in [0, 1($: unendlicher String zufälliger Bits
Idee: u und v tauschen nur die ersten $(c + 3)\lceil \log n \rceil$ Bits aus

Wahrscheinlichkeit, dass die ersten $(c + 3)\lceil \log n \rceil$ Bits gleich sind:

$$\left(\frac{1}{2}\right)^{(c+3)\lceil \log n \rceil} \leq \left(\frac{1}{2}\right)^{(c+3) \log n} = \frac{1}{n^{c+3}}$$

Bandbreitenbeschränkung

Ziel: Für jede Kante $\{u, v\}$, entscheide, ob $r(u) < r(v)$ oder $r(u) > r(v)$

Zufällige reelle Zahl $\in [0, 1($: unendlicher String zufälliger Bits
Idee: u und v tauschen nur die ersten $(c + 3)\lceil \log n \rceil$ Bits aus

Wahrscheinlichkeit, dass die ersten $(c + 3)\lceil \log n \rceil$ Bits gleich sind:

$$\left(\frac{1}{2}\right)^{(c+3)\lceil \log n \rceil} \leq \left(\frac{1}{2}\right)^{(c+3)\log n} = \frac{1}{n^{c+3}}$$

Union Bound:
Wahrscheinlichkeit, dass für irgendeines der $\leq n^2$ Paare in den $\leq n$ Phasen des Algorithmus die ersten $(c + 3)\lceil \log n \rceil$ Bits gleich sind: $\leq n^3 \cdot \frac{1}{n^{c+3}} = \frac{1}{n^c}$

Bandbreitenbeschränkung

Ziel: Für jede Kante $\{u, v\}$, entscheide, ob $r(u) < r(v)$ oder $r(u) > r(v)$

Zufällige reelle Zahl $\in [0, 1($: unendlicher String zufälliger Bits
Idee: u und v tauschen nur die ersten $(c+3)\lceil \log n \rceil$ Bits aus

Wahrscheinlichkeit, dass die ersten $(c+3)\lceil \log n \rceil$ Bits gleich sind:

$$\left(\frac{1}{2}\right)^{(c+3)\lceil \log n \rceil} \leq \left(\frac{1}{2}\right)^{(c+3)\log n} = \frac{1}{n^{c+3}}$$

Union Bound:
Wahrscheinlichkeit, dass für irgendeines der $\leq n^2$ Paare in den $\leq n$ Phasen des Algorithmus die ersten $(c+3)\lceil \log n \rceil$ Bits gleich sind: $\leq n^3 \cdot \frac{1}{n^{c+3}} = \frac{1}{n^c}$

Somit: Bandbreite $O(\log n)$ ist ausreichend

Robustheit der „hohen Wahrscheinlichkeit“

Definition

Ein Ereignis findet **mit hoher Wahrscheinlichkeit** statt, wenn es mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$, für eine beliebig vorgegebene Konstante $c \geq 1$, stattfindet.

Robustheit der „hohen Wahrscheinlichkeit“

Definition

Ein Ereignis findet **mit hoher Wahrscheinlichkeit** statt, wenn es mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$, für eine beliebig vorgegebene Konstante $c \geq 1$, stattfindet.

$\rightarrow$ In der Regel hängt die in der O-Notation verborgene Konstante von c ab

Robustheit der „hohen Wahrscheinlichkeit“

Definition

Ein Ereignis findet **mit hoher Wahrscheinlichkeit** statt, wenn es mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$, für eine beliebig vorgegebene Konstante $c \geq 1$, stattfindet.

$\rightarrow$ In der Regel hängt die in der O-Notation verborgene Konstante von c ab

Beispiel: n Ereignisse $A_1, \ldots A_n$, die jeweils mit hoher Wahrscheinlichkeit eintreten; Wie wahrscheinlich ist es, dass alle gleichzeitig eintreten?

Robustheit der „hohen Wahrscheinlichkeit"

Definition

Ein Ereignis findet **mit hoher Wahrscheinlichkeit** statt, wenn es mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$, für eine beliebig vorgegebene Konstante $c \geq 1$, stattfindet.

$\longrightarrow$ In der Regel hängt die in der O-Notation verborgene Konstante von c ab

Beispiel: n Ereignisse $A_1, \ldots A_n$, die jeweils mit hoher Wahrscheinlichkeit eintreten; Wie wahrscheinlich ist es, dass alle gleichzeitig eintreten?

$$\Pr\left[\bigcap_{i=1}^{n} A_i\right] = 1 - \Pr\left[\bigcup_{i=1}^{n} \bar{A}_i\right]$$

Robustheit der „hohen Wahrscheinlichkeit“

Definition

Ein Ereignis findet **mit hoher Wahrscheinlichkeit** statt, wenn es mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$, für eine beliebig vorgegebene Konstante $c \geq 1$, stattfindet.

$\rightarrow$ In der Regel hängt die in der O-Notation verborgene Konstante von c ab

Beispiel: n Ereignisse $A_1, \ldots A_n$, die jeweils mit hoher Wahrscheinlichkeit eintreten; Wie wahrscheinlich ist es, dass alle gleichzeitig eintreten?

$$\Pr\left[\bigcap_{i=1}^{n} A_i\right] = 1 - \Pr\left[\bigcup_{i=1}^{n} \bar{A}_i\right] \geq 1 - \sum_{i=1}^{n} \Pr([\bar{A}_i]$$

Robustheit der „hohen Wahrscheinlichkeit“

Definition

Ein Ereignis findet **mit hoher Wahrscheinlichkeit** statt, wenn es mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$, für eine beliebig vorgegebene Konstante $c \geq 1$, stattfindet.

$\rightarrow$ In der Regel hängt die in der O-Notation verborgene Konstante von c ab

Beispiel: n Ereignisse $A_1, \ldots A_n$, die jeweils mit hoher Wahrscheinlichkeit eintreten; Wie wahrscheinlich ist es, dass alle gleichzeitig eintreten?

$$\Pr\left[\bigcap_{i=1}^{n} A_i\right] = 1 - \Pr\left[\bigcup_{i=1}^{n} \bar{A}_i\right] \geq 1 - \sum_{i=1}^{n} \Pr([\bar{A}_i] = 1 - \sum_{i=1}^{n} (1 - \Pr[A_i])$$

Robustheit der „hohen Wahrscheinlichkeit“

Definition

Ein Ereignis findet **mit hoher Wahrscheinlichkeit** statt, wenn es mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$, für eine beliebig vorgegebene Konstante $c \geq 1$, stattfindet.

$\rightarrow$ In der Regel hängt die in der O-Notation verborgene Konstante von c ab

Beispiel: n Ereignisse $A_1, \ldots A_n$, die jeweils mit hoher Wahrscheinlichkeit eintreten; Wie wahrscheinlich ist es, dass alle gleichzeitig eintreten?

$$\Pr\left[\bigcap_{i=1}^{n} A_i\right] = 1 - \Pr\left[\bigcup_{i=1}^{n} \bar{A}_i\right] \geq 1 - \sum_{i=1}^{n} \Pr([\bar{A}_i] = 1 - \sum_{i=1}^{n} (1 - \Pr[A_i])$$

$$\geq 1 - \sum_{i=1}^{n} \frac{1}{n^c} = 1 - \frac{n}{n^c} = 1 - \frac{1}{n^{c-1}}$$

Robustheit der „hohen Wahrscheinlichkeit“

Definition

Ein Ereignis findet **mit hoher Wahrscheinlichkeit** statt, wenn es mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$, für eine beliebig vorgegebene Konstante $c \geq 1$, stattfindet.

→ In der Regel hängt die in der O-Notation verborgene Konstante von c ab

Beispiel: n Ereignisse $A_1, \ldots A_n$, die jeweils mit hoher Wahrscheinlichkeit eintreten; Wie wahrscheinlich ist es, dass alle gleichzeitig eintreten?

$$\Pr\left[\bigcap_{i=1}^{n} A_i\right] = 1 - \Pr\left[\bigcup_{i=1}^{n} \bar{A}_i\right] \geq 1 - \sum_{i=1}^{n} \Pr([\bar{A}_i] = 1 - \sum_{i=1}^{n}(1 - \Pr[A_i])$$

$$\geq 1 - \sum_{i=1}^{n} \frac{1}{n^c} = 1 - \frac{n}{n^c} = 1 - \frac{1}{n^{c-1}}$$

Wenn die Wahrscheinlichkeit für jedes A_i auf $1 - \frac{1}{n^{c+1}}$ erhöht werden kann, dann findet $\left(\bigcap_{i=1}^{n} A_i\right)$ „mit hoher Wahrscheinlichkeit“ statt.

Zusammenfassung

Maximal Independent Set

- Prototypisches „lokales“ Problem
- Einfache sequentielle Lösung

Zusammenfassung

Maximal Independent Set

- Prototypisches „lokales“ Problem
- Einfache sequentielle Lösung
- Randomisierter Algorithmus zur „parallelen“ Vergrößerung des MIS ohne Koordination
- Analyse mit typische Werkzeugen randomisierter Algorithmen
- „Idealer“ Algorithmus wird mit beschränkter Bandbreite implementiert

Quellen

Der Inhalt dieser Vorlesungseinheit basiert zum Teil auf Vorlesungseinheiten von Christoph Lenzen und Roger Wattenhofer.

Literatur:

- Noga Alon, László Babai, Alon Itai. „A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem". *Journal of Algorithms* 7(4): 567–583 (1986)
- Michael Luby. „A Simple Parallel Algorithm for the Maximal Independent Set Problem". *SIAM Journal on Computing* 15(4): 1036–1053 (1986)
- Yves Métivier, John Michael Robson, Nasser Saheb-Djahromi, Akka Zemmari. „An optimal bit complexity randomized distributed MIS algorithm". *Distributed Computing* 23(5–6): 331–340 (2011)