1. Cuckoo hashing with more than two hash functions

2. The Peeling Algorithm

3. The Peeling Theorem
1. Cuckoo hashing with more than two hash functions

2. The Peeling Algorithm

3. The Peeling Theorem
Cuckoo Hashing with one table and $k$ hash functions

$n \in \mathbb{N}$ keys
$m \in \mathbb{N}$ table size
$\alpha = \frac{n}{m}$ load factor
$h_1, \ldots, h_k \sim \mathcal{U}([m])$ hash functions

$\leftrightarrow$ Could also use a separate table per hash function.

**Theorem (without proof)**

For each $k \in \mathbb{N}$ there is a threshold $c_k^*$ such that:
- if $\alpha < c_k^*$ all keys can be placed with probability $1 - \mathcal{O}(\frac{1}{m})$.
- if $\alpha > c_k^*$ not all keys can be placed with probability $1 - \mathcal{O}(\frac{1}{m})$.

$c_2^* = \frac{1}{2}, \quad c_3^* \approx 0.92, \quad c_4^* \approx 0.98, \ldots$

**Conjecture**

If $\alpha < c_k^*$ then the expected number of steps of successful insertions is $\mathcal{O}(1)$.

$\leftrightarrow$ several proof attempts for random walk and other algorithms exist, with partial success.

---

```
randomWalkInsert(x)
while x ≠ ⊥ do // TODO: limit
    sample i ~ U([k])
    swap(x, T[h_i(x)])

(some improvements possible)
```

---

Cuckoo hashing with more than two hash functions

Cuckoo hashing with more than two hash functions

The Peeling Algorithm

The Peeling Theorem

ITI, Algorithm Engineering & Scalable Algorithms
Static Hash Tables

**Static Hash Table**

- **construct(S):** builds table $T$ with key set $S$
- **lookup(x):** checks if $x$ is in $T$ or not
- $\implies$ no insertions or deletions after construction!

**Constructing cuckoo hash tables:**

- solved by Khosla 2013: “Balls into Bins Made Faster”
- matching algorithm resembling preflow push
- expected running time $O(n)$, finds placement whenever one exists
- not in this lecture

**Greedyly constructing cuckoo hash tables**

- Peeling algorithm: simple but sophisticated analysis
- interesting applications beyond hash tables (see “retrieval” in next lecture)
Content

1. Cuckoo hashing with more than two hash functions

2. The Peeling Algorithm

3. The Peeling Theorem
The Peeling Algorithm

constructByPeeling\((S \subseteq D, h_1, h_2, h_3 \in [m]^D)\)

\[
T \leftarrow [\bot, \ldots, \bot] \quad // \text{empty table of size } m
\]

\[
\text{while } \exists i \in [m] : \exists \text{ exactly one } x \in S : i \in \{h_1(x), h_2(x), h_3(x)\} \text{ do}
\]

\[
\quad // x \text{ is only unplaced key that may be placed in } i
\quad T[i] \leftarrow x
\quad S \leftarrow S \setminus \{x\}
\]

\[
\quad \text{if } S = \emptyset \text{ then}
\quad \quad \text{return } T
\]

\[
\quad \text{else}
\quad \quad \text{return NOT-PEELABLE}
\]

Exercise

- Success of constructByPeeling does not depend on choices for \(i\) made by while.
- constructByPeeling can be implemented in linear time.
Peelability and the Cuckoo Graph

Cuckoo Graph and Peelability

- The **Cuckoo Graph** is the bipartite graph
  \[ G_{S,h_1,h_2,h_3} = (S, [m], \{(x, h_i(x)) \mid x \in S, i \in [3]\}) \]

- Call \( G_{S,h_1,h_2,h_3} \) **peelable** if \( \text{constructByPeeling}(S, h_1, h_2, h_3) \) succeeds.

- If \( h_1, h_2, h_3 \sim \mathcal{U}([m]^D) \) then the distribution of \( G_{S,h_1,h_2,h_3} \) does not depend on \( S \). We then simply write \( G_{m,\alpha m} \).
  - \( m \square \)-nodes and \( \lfloor \alpha m \rfloor \)-nodes
  - think: \( \alpha \) is constant and \( m \to \infty \).

Peeling simplified (not computing placement)

\[
\text{while } \exists \square\text{-node of degree 1 do} \quad \text{G is peelable if and only if this algorithm removes all } \bigcirc \text{-nodes.}
\]

while \( \exists \square\)-node of degree 1 do

\[ \text{remove it and its incident } \bigcirc \]
1. Cuckoo hashing with more than two hash functions

2. The Peeling Algorithm

3. The Peeling Theorem
Peeling Threshold

Let \( c_3^\Delta = \min_{y \in [0,1]} \frac{y}{3(1-e^{-y})^2} \approx 0.81 \).

Theorem (today’s goal)

Let \( \alpha < c_3^\Delta \). Then \( \Pr[G_{m,\alpha m} \text{ is peelable}] = 1 - o(1) \).

Remark: More is known.

- For \( \alpha < c_3^\Delta \) we get “peelable” with probability \( 1 - O(1/m) \).
- For \( \alpha > c_3^\Delta \) we get “not peelable” with probability \( 1 - O(1/m) \).
- Corresponding thresholds \( c_k^\Delta \) for \( k \geq 3 \) hash functions are also known.

Exercise: What about \( k = 2 \)?

Peeling does not reliably work for \( k = 2 \) for any \( \alpha > 0 \).
Theorem (today’s goal)

Let $\alpha < c^3_3$. Then $\Pr[G_{m,\alpha} m \text{ is peelable}] = 1 - o(1)$.

Proof Idea

The random (possibly) infinite tree $T_\alpha$ can be peeled for $\alpha < c^3_3$ and $T_\alpha$ is locally like $G_{m,\alpha} m$.

Steps

I. What is an infinite tree in general?
II. What is $T_\alpha$ in particular?
III. What does peeling mean in this setting?
IV. What role does $c^3_3$ play?
V. What does it mean for $T_\alpha$ to be locally like $G_{m,\alpha} m$?
VI. What is the probability that a fixed key of $G_{m,\alpha} m$ is peeled?
VII. What is the probability that all keys of $G_{m,\alpha} m$ are peeled?
What is an infinite tree in general?

Tree Definitions

- connected and acyclic ✓
  sensible and satisfied
- connected and $|E| = |V| - 1$ ✗
  not sensible

$V = \mathbb{N}$
$E = \{\{n, 2n\} | n \in \mathbb{N}\} \cup \{\{n, 2n + 1\} | n \in \mathbb{N}\}$. 

Cuckoo hashing with more than two hash functions

The Peeling Algorithm

The Peeling Theorem
Observations for the finite Graph $G_{m,\alpha m}$

- each $\bigcirc$ has 3 $\Box$ as neighbours (rare exception: $h_1(x), h_2(x), h_3(x)$ not distinct)
- each $\Box$ has random number $X$ of $\bigcirc$ as neighbours with $X \sim \text{Bin}(3n, \frac{1}{m}) = \text{Bin}(3\lfloor \alpha m \rfloor, \frac{1}{m})$. In an exercise you’ll show

$$\Pr[X = i] \xrightarrow{m \to \infty} \Pr_{Y \sim \text{Pois}(3\alpha)}[Y = i].$$

Definition of the (possibly) infinite random tree $T_\alpha$

- root is $\bigcirc$ and has three $\Box$ as children
- each $\Box$ has random number of $\bigcirc$ children, sampled $\text{Pois}(3\alpha)$ (independently for each $\Box$).
- each non-root $\bigcirc$ has two $\Box$ as children.

Remark: $T_\alpha$ is finite with positive probability $> 0$, e.g. when the first three $\text{Pois}(3\alpha)$ random variables come out as 0. But $T_\alpha$ is also infinite with positive probability.
What does peeling mean in this setting?

**Peeling Algorithm**

while ∃ childless □-node do
  remove it and its incident ○

→ not well defined outcome on $T_\alpha$!
→ but well defined on $T_\alpha^R$!

**Peel only the first $R \in \mathbb{N}$ layers**

- Let $T_\alpha^R$ be the first $2R + 1$ levels of $T_\alpha$.
- $R$ layers of □-nodes, labeled bottom to top.
- Run peeling on $T_\alpha^R$ (later $R \to \infty$).

→ Why not consider the first $2R$ levels? (without +1)

**Only care whether root is removed (root represents arbitrary node in $G_{m,\alpha,m}$)**

We may then simplify the peeling algorithm.

- replace “□-node of degree 1” condition with stronger “childless □-node”.
  - prevents peeling of □-nodes with one child and no parent
  - no matter: such nodes are disconnected from the root anyway
- whether node is peeled only depends on subtree
  → one bottom up pass suffices for peeling
**Observation**

Let \( q_R = \Pr[\text{root survives when peeling } T^R_\alpha] \).

The values \( q_R \) are decreasing in \( R \).

---

**Proof.**

Assume when peeling \( T^R_\alpha \) the sequence \( \vec{x} = (x_1, \ldots, x_k) \) is a valid sequence of \( \Box \)-node choices. Then \( \vec{x} \) is also valid when peeling \( T^{R+1}_\alpha \).

- Peeling \( T^R_\alpha \) removes the root \( \Rightarrow \) peeling \( T^{R+1}_\alpha \) removes the root
- Root survives when peeling \( T^{R+1}_\alpha \) \( \Rightarrow \) peeling \( T^R_\alpha \) removes the root
- \( q_{R+1} \leq q_R \)

---

**Peeling Algorithm**

```plaintext
while \( \exists \) childless \( \Box \)-node do
  remove it and its incident
```

---

The Peeling Algorithm
iii What does peeling mean in this setting? (3)

Peeling $T^R$ bottom up

for $i = 1$ to $R$ do // layers bottom to top
    for each □-node $v$ in layer $i$ do
        if $v$ has no children then
            remove $v$ and its parent ○

Survival probabilities $p_i := \Pr[□$-node in layer $i$ is not peeled]

\[
p_1 = \Pr[□$-node has $\geq 1$ child] = \Pr_{Y \sim \text{Pois}(3\alpha)}[Y > 0] = 1 - e^{-3\alpha}.
\]
\[
p_i = \Pr[\text{layer } i \text{ □-node } v \text{ has } \geq 1 \text{ surviving child}] = \Pr_{X \sim \text{Pois}(3\alpha p^2_{i-1})}[X > 0] = 1 - e^{-3\alpha p^2_{i-1}}.
\]

$Y := \text{number of (initial) children of } v$

$X := \text{number of surviving children of } v$ each child □-node survives if both its □-children from layer $i - 1$ survive $\rightsquigarrow$ probability $p^2_{i-1}$.

$\Rightarrow Y \sim \text{Pois}(3\alpha)$ and $X \sim \text{Bin}(Y, p^2_{i-1})$.

$\Rightarrow X \sim \text{Pois}(3\alpha p^2_{i-1})$. $\rightsquigarrow$ exercise!
What does peeling mean in this setting? (3)

Peeling $T^R_\alpha$ bottom up

for $i = 1$ to $R$ do // layers bottom to top
  for each $\square$-node $v$ in layer $i$ do
    if $v$ has no children then
      remove $v$ and its parent

Survival probabilities $p_i := \Pr[\square$-node in layer $i$ is not peeled]

\[
p_1 = \Pr[\square$-node has $\geq 1$ child] = \Pr[Y \sim \text{Pois}(3\alpha)] [Y > 0] = 1 - e^{-3\alpha}.
\]
\[
p_i = \Pr[\text{layer } i \square$-node $v$ has $\geq 1$ surviving child] = \Pr[X \sim \text{Pois}(3\alpha p^2_{i-1})] [X > 0] = 1 - e^{-3\alpha p^2_{i-1}}.
\]

- survival probabilities. With $p_0 := 1$ we have
  \[
p_i = \begin{cases} 1 & \text{if } i = 0 \\ 1 - e^{-3\alpha p^2_{i-1}} & \text{if } i = 1, 2, \ldots. \end{cases}
\]

Moreover: $q_R := \Pr[\text{root survives}] = p^3_R$. 

Cuckoo hashing with more than two hash functions

The Peeling Algorithm

The Peeling Theorem
What role does $c_3^\Delta \approx 0.81$ play?

$$p_i = \begin{cases} 
1 & \text{if } i = 0 \\
1 - e^{-3\alpha p_{i-1}^2} & \text{if } i = 1, 2, \ldots 
\end{cases}$$

→ consider $f(x) = 1 - e^{-3\alpha x^2}$

**Case 1:** $\exists x > 0 : f(x) = x$.  

$\Rightarrow \lim_{i \to \infty} p_i = x^* = \max \{x \in [0,1] \mid f(x) = x\}$.

**Case 2:** $\forall x \in (0,1) : f(x) < x$.  

$\Rightarrow \lim_{i \to \infty} p_i = 0$. 
What role does $c_3^\Delta \approx 0.81$ play?

$p_i = \begin{cases} 1 & \text{if } i = 0 \\ 1 - e^{-3\alpha p_{i-1}} & \text{if } i = 1, 2, \ldots \end{cases}$

$\leftarrow$ consider $f(x) = 1 - e^{-3\alpha x^2}$

Case 1: $\exists x > 0 : f(x) = x$.

$\lim_{i \to \infty} p_i = x^* = \max\{x \in [0, 1] \mid f(x) = x\}.$

$\Rightarrow \exists z > 0 : \alpha = \frac{z}{3(1 - e^{-z})^2}$

$\Rightarrow \alpha \geq \min_{z > 0} \frac{z}{3(1 - e^{-z})^2}$

$\Delta_3 \approx 0.81$
Lemma

For $\alpha < c_3^\Delta \approx 0.81$ we have

1. $\lim_{i \to \infty} p_i = 0$.
2. $\lim_{R \to \infty} q_R = \lim_{R \to \infty} p_R^3 = 0$.

“Root rarely survives for large $R$.”

“Root rarely survives for large $R$.”
What does it mean for $T_\alpha$ to be locally like $G_{m,\alpha m}$?

**Neighbourhoods in $T_\alpha$ and $G$**

Let $R \in \mathbb{N}$. We consider

- $T_\alpha^R$ as before and
- for any fixed $x \in S$ the subgraph $G_{m,\alpha m}^{x,R}$ of $G_{m,\alpha m}$ induced by all nodes with distance at most $2R$ from $x$.

**Lemma**

For any $R \in \mathbb{N}$, the distribution of $G_{m,\alpha m}^{x,R}$ converges the distribution of $T_\alpha^R$, i.e.

$$\forall T : \lim_{m \to \infty} \Pr[G_{m,\alpha m}^{x,R} = T] = \Pr[T_\alpha^R = T].$$

Cuckoo hashing with more than two hash functions

The Peeling Algorithm

The Peeling Theorem
Lemma

Let \( T_y \) be a possible outcome of \( T^R_\alpha \) given by a finite sequence \( y = (y_1, \ldots, y_k) \in \mathbb{N}_0^k \) specifying the number of children of \( \Box \)-nodes in level order. Then

\[
\Pr[\ T^R_\alpha = T_y \ ] = \prod_{i=1}^{k} \Pr_{Y \sim \text{Pois}(3\alpha)}[Y = y_i].
\]

E.g. for \( y = (2, 0, 1, 4, 2, 1, 0, 3, 2) \):
Lemma

Assume \( R = \mathcal{O}(1) \). The probability that \( G_{m, \alpha m}^{x, R} \) contains a cycle is \( \mathcal{O}(1/m) \).

Proof.

If \( G_{m, \alpha m}^{x, R} \) contains a cycle then we have

- a sequence \( (v_1 = x, v_2, \ldots, v_k, v_{k+1} = v_a) \) of nodes with \( a \in [k] \)
- of length \( k \leq 4R \) (consider BFS tree for \( x \) and additional edge in it)
- for each \( i \in \{1, \ldots, k\} \) an index \( j_i \in \{1, 2, 3\} \) of the hash function connecting \( v_i \) and \( v_{i+1} \). (If \( a = k - 1 \) then \( j_k = j_{k-1} \).)

\[
\Pr[\exists \text{cycle in } G_{m, \alpha m}^{x, R}] \leq \Pr[\exists 2 \leq k \leq 4R : \exists v_2, \ldots, v_k : \exists a \in [k] : \exists j_1, \ldots, j_k \in [3] : \forall i \in [k] : h_{j_i} \text{ connects } v_i \text{ to } v_{i+1}]
\]

\[
\leq \sum_{k=2}^{4R} \sum_{v_2, \ldots, v_k} \sum_{a=1}^k \sum_{j_1, \ldots, j_k} \prod_{i=1}^k \Pr[h_{j_i} \text{ connects } v_i \text{ to } v_{i+1}] \leq \sum_{k=2}^{4R} \left( \max\{m, n\} \right)^{k-1} \cdot k \cdot 3^k \left( \frac{1}{m} \right)^k = \frac{1}{m} \sum_{k=2}^{4R} k \cdot 3^k = \mathcal{O}(1/m). \quad \square
\]
Lemma

Let $T_y$ be a possible outcome of $T^R_\alpha$ as before. Then

$$\Pr_{h_1, h_2, h_3 \sim U([m]^D)}[G^x, R_{m, \alpha m} = T_y] \xrightarrow{m \to \infty} \prod_{i=1}^{k} \Pr_{Y \sim \text{Pois}(3\alpha)}[Y = y_i].$$

"Proof by example", using $T_y$ shown on the right.

The following things have to "go right" for $G^x, R_{m, \alpha m} = T_y$.

- $h_1(x), h_2(x), h_3(x)$ pairwise distinct: probability $\xrightarrow{m \to \infty} 1$
  $\iff$ non-distinct would give cycle of length 2. Unlikely by lemma.

Note: $3 \lfloor \alpha m \rfloor - 3$ remaining hash values $\sim U([m])$. 

---

Cuckoo hashing with more than two hash functions

The Peeling Algorithm

The Peeling Theorem
**Distribution of** $G_{m,\alpha m}^{x,R}$

**Lemma**

Let $T_y$ be a possible outcome of $T^R_{\alpha}$ as before. Then

$$\Pr_{h_1, h_2, h_3 \sim \mathcal{U}([m])}[G_{m,\alpha m}^{x,R} = T_y] \xrightarrow{m \to \infty} \prod_{i=1}^{k} \Pr_{Y \sim \text{Pois}(3\alpha)}[Y = y_i].$$

**“Proof by example”, using $T_y$ shown on the right.**

- Exactly $y_1 = 2$ of the remaining hash values are $u$.

  $$\Pr_{Y \sim \text{Bin}(3 \lfloor \alpha m \rfloor - 3, \frac{1}{m})}[Y = 2] \xrightarrow{m \to \infty} \Pr_{Y \sim \text{Pois}(3\alpha)}[Y = 2]. \xrightarrow{\text{exercise}}$$

  Moreover: The two hash values must belong to 2 distinct keys. Probability $\xrightarrow{m \to \infty} 1$.

  $$\xrightarrow{\text{exercise}}$$

  non-distinct would give cycle of length 2.

**Note:** The $3 \lfloor \alpha m \rfloor - 5$ remaining hash values are $\sim \mathcal{U}([m] \setminus \{u\})$. \xrightarrow{\text{exercise}}
Lemma

Let $T_y$ be a possible outcome of $T^R_\alpha$ as before. Then

$$\Pr_{h_1, h_2, h_3 \sim \mathcal{U}([m]^D)}[G^{x, R}_{m, \alpha m} = T_y] \xrightarrow{m \to \infty} \prod_{i=1}^{k} \Pr_{Y \sim \text{Pois}(3\alpha)}[Y = y_i].$$

“Proof by example”, using $T_y$ shown on the right.

- None of the remaining hash values are $v$.
  $$\leftrightarrow \Pr_{Y \sim \text{Bin}(3\lfloor \alpha m \rfloor - 5, \frac{1}{m-1})}[Y = 0] \xrightarrow{m \to \infty} \Pr_{Y \sim \text{Pois}(3\alpha)}[Y = 0].$$
  Note: The $3\lfloor \alpha m \rfloor - 5$ remaining hash values are $\sim \mathcal{U}([m] \setminus \{u, v\})$.

- One of the remaining hash values is $w$.
  $$\leftrightarrow \Pr_{Y \sim \text{Bin}(3\lfloor \alpha m \rfloor - 5, \frac{1}{m-2})}[Y = 1] \xrightarrow{m \to \infty} \Pr_{Y \sim \text{Pois}(3\alpha)}[Y = 1].$$
  ...
Lemma

Let $T_y$ be a possible outcome of $T_R^\alpha$ as before. Then

$$\Pr_{h_1, h_2, h_3 \sim \mathcal{U}([m]^D)}[G_{m, \alpha m}^{x, R} = T_y] \xrightarrow{m \to \infty} \prod_{i=1}^{k} \Pr_{Y \sim \text{Pois}(3\alpha)}[Y = y_i].$$

Proof sketch in general (some details ommitted)

- General case at $i$-th □-node. Want: probability that $G_{m, \alpha m}^{x, R}$ continues to match $T_y$. Note: $T_y$ is fixed, so $i$ and the number $c_i$ of previously revealed hash values is bounded.

$$\Pr_{Y \sim \text{Bin}(3 \lfloor \alpha m \rfloor - c_i, \frac{1}{m-i+1})}[Y = y_i] \xrightarrow{m \to \infty} \Pr_{Y \sim \text{Pois}(3\alpha)}[Y = y_i].$$

Moreover, those $y_i$ hash values must belong to distinct fresh keys. Probability $\xrightarrow{m \to \infty} 1$ ↔ otherwise we’d have a cycle.

- General case for ○-node. The two children must be fresh: probability $\xrightarrow{m \to \infty} 1$ ↔ otherwise there would be a cycle.
Lemma

Let $\alpha < c_3^\Delta$. Let $x$ be any $\bigcirc$-node in $G_{m,\alpha m}$ as before (chosen before sampling the hash functions). Let

$$\mu_m := \Pr_{h_1, h_2, h_3 \sim \mathcal{U}([m]^\alpha)}[x \text{ is removed when peeling } G_{m,\alpha m}].$$

Then $\lim_{m \to \infty} \mu_m = 1$. 

Probability that a specific key survives peeling
Let $\delta > 0$ be arbitrary. We will show $\lim_{m \to \infty} \mu_m \geq 1 - 2\delta$.

Let $R \in \mathbb{N}$ be such that $q_R < \delta$.

$\mathcal{Y}_R^a := \{T \in \mathcal{Y}_R | \text{ peeling } T \text{ removes the root}\}$

Let $\mathcal{Y}_R^a \subseteq \mathcal{Y}_R$ be a finite set such that $\Pr[T^a_\alpha / \in \mathcal{Y}_R^a] \leq \delta$

$$\lim_{m \to \infty} \mu_m \geq \lim_{m \to \infty} \Pr[G_{m, \alpha m}^x \in \mathcal{Y}_R^\text{peel}] \geq \lim_{m \to \infty} \Pr[G_{m, \alpha m}^x \in \mathcal{Y}_R^\text{peel} \cap \mathcal{Y}_R^\text{fin}] = \lim_{m \to \infty} \sum_{T \in \mathcal{Y}_R^\text{peel} \cap \mathcal{Y}_R^\text{fin}} \Pr[G_{m, \alpha m}^x = T] = \sum_{T \in \mathcal{Y}_R^\text{peel} \cap \mathcal{Y}_R^\text{fin}} \lim_{m \to \infty} \Pr[G_{m, \alpha m}^x = T] = \sum_{T \in \mathcal{Y}_R^\text{peel} \cap \mathcal{Y}_R^\text{fin}} \Pr[T^a_\alpha = T] = \Pr[T^a_\alpha \in \mathcal{Y}_R^\text{peel} \cap \mathcal{Y}_R^\text{fin}] = 1 - \Pr[T^a_\alpha \notin \mathcal{Y}_R^\text{peel} \cap \mathcal{Y}_R^\text{fin}] \geq 1 - \Pr[T^a_\alpha \notin \mathcal{Y}_R^\text{peel}] - \Pr[T^a_\alpha \notin \mathcal{Y}_R^\text{fin}] \geq 1 - 2\delta.$
Proof of the Peeling Theorem

Theorem

Let $\alpha < c_3^\Delta$. Then

$$\Pr[G_{m,\alpha m} \text{ is peelable}] = 1 - o(1).$$

Proof

Let $n = \lfloor \alpha m \rfloor$ and $0 \leq s \leq n$ the number of $\bigcirc$ nodes surviving peeling.

last lemma: each $\bigcirc$ survives with probability $o(1)$.

linearity of expectation \[ E[s] = n \cdot o(1) = o(n). \]

Exercise: \[ \Pr[s \in \{1, \ldots, \delta n\}] = \mathcal{O}(1/m) \text{ if } \delta > 0 \text{ is a small enough constant.} \]

Markov: \[ \Pr[s > \delta n] \leq \frac{E[s]}{\delta n} = \frac{o(n)}{\delta n} = o(1). \]

finally: \[ \Pr[s > 0] = \Pr[s \in \{1, \ldots, \delta n\}] + \Pr[s > \delta n] = \mathcal{O}(1/m) + o(1) = o(1). \]
Peeling Process
- greedy algorithm for placing keys in cuckoo table
- works up to a load factor of $c_3^\Delta \approx 0.81$

We saw glimpses of important techniques
- Local interactions in large graphs. Also used in statistical physics.
- Local weak convergence. How the finite graph $G_{m,\alpha,m}$ is locally like $T_\alpha$.

But wait, there’s more!
- Further applications of peeling
  - retrieval data structures (next lecture)
  - perfect hash functions (next lecture)
- set sketches
- linear error correcting codes

Conclusion
Cuckoo Hashing und der Schälalgorithmus

- (Wie) kann man Cuckoo Hashing mit mehr als 2 Hashfunktionen aufziehen?
- Welcher Vorteil ergibt sich im Vergleich zu 2 Hashfunktionen?
- Wie funktioniert der Schälalgorithmus zur Platzierung von Schlüsseln in einer Cuckoo Hashtabelle?
- Schälen lässt sich als einfacher Prozess auf Graphen auffassen. Wie?
- Was besagt das Hauptresultat, das wir zum Schälprozess bewiesen haben?

Beweis des Schälsatzes. *Mir ist klar, dass der Beweis äußerst kompliziert ist.*

- Im Beweis haben zwei Graphen eine Rolle gespielt ein endlicher und ein (potentiell) unendlicher. Wie waren diese Graphen definiert?
- Welcher Zusammenhang besteht zwischen der Verteilung der Knotengrade in $T_\alpha$ und $G_{m,\alpha,m}$?