Homework 2 Released
Project Proposals due February 21

Advanced Cryptography
CS 655

Week 6:

- Memory Hard Functions and Pebbling
- Pebbling Attacks
- Depth-Robust Graphs and Pebbling
- Constructing Depth-Robust Graphs

Course Project Proposal

- You may complete your project individually or as a group of size two.
- You are welcome to come up with your own project or talk to me for ideas.
- Project Proposal: 2 Pages
  - Briefly the problem you plan to work on
  - Briefly summarize prior work on the problem and how your project is different
  - Identify several related papers that you plan to read as part of the project
  - Briefly describe your plan to attack the problem

A Few Project Ideas

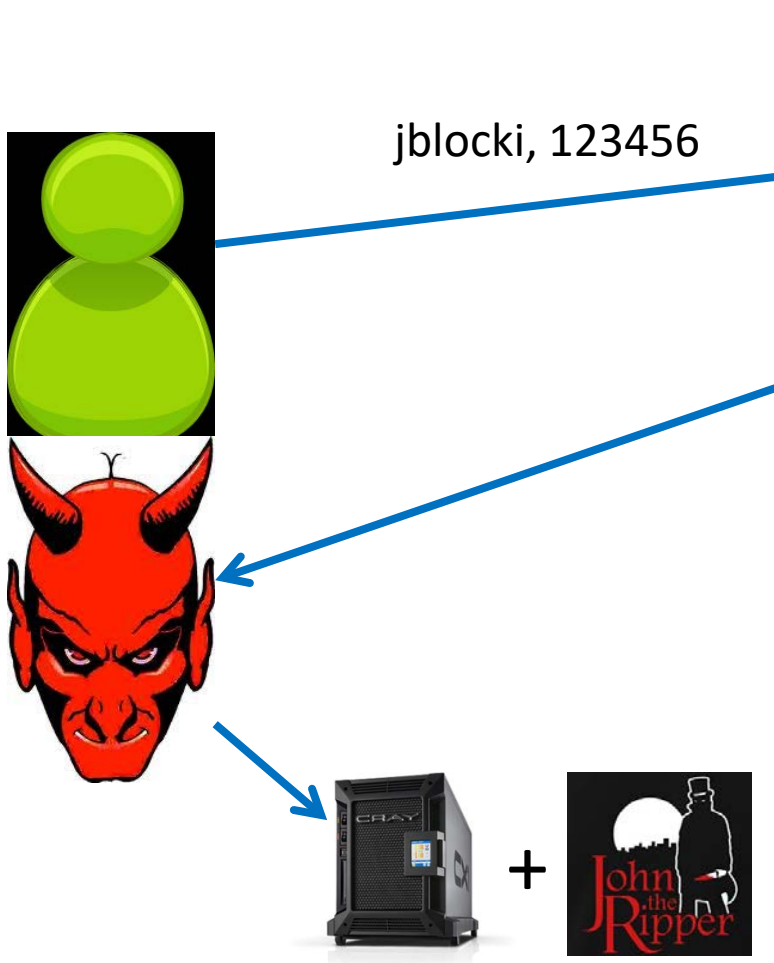
- Pick a cryptographic scheme and try to find a tighter concrete security proof under idealized assumptions
  - Example: Tighter security analysis for Password Authenticated Key Exchange (PAKE) protocols such as CPACE in the generic group+random oracle model?
- Pick a cryptographic scheme/protocol and analyze the security with respect pre-processing attacks or provide a memory-tight reduction
  - **Example:** Memory-Tight Reduction for RSA-FDH under the One-More-RSA-Inversion problem?
  - **Example:** Security of PAKE protocols against pre-processing attacks?
  - **Example:** Security of AES-GCM vs pre-processing attacks?
- Pebbling Reduction for Salted iMHFs vs. Preprocessing Attackers
- Pebbling Reduction for Argon2 Round Function (in ideal permutation model)

A Few Project Ideas

- Implement a Cryptographic Protocol/Attack
  - **Example:** Implement Argon2 with different instantiations of round function
  - **Example:** Implement partitioning oracle attack on AES-GCM.
- Many other possibilities! Make sure your proposal is realistic.

- It is ok to try something and fail i.e., a final project report documenting your unsuccessful attempts to solve a problem is acceptable as long as the attempts are clearly described

Motivation: Password Storage



Username	Salt	Hash
jblocki	89d978034a3f6	85e23cfe0021f584e3db87aa72630a9a2345c062

SHA1(12345689d978034a3f6)=85e23cfe0021f584e3db87aa72630a9a2345c062

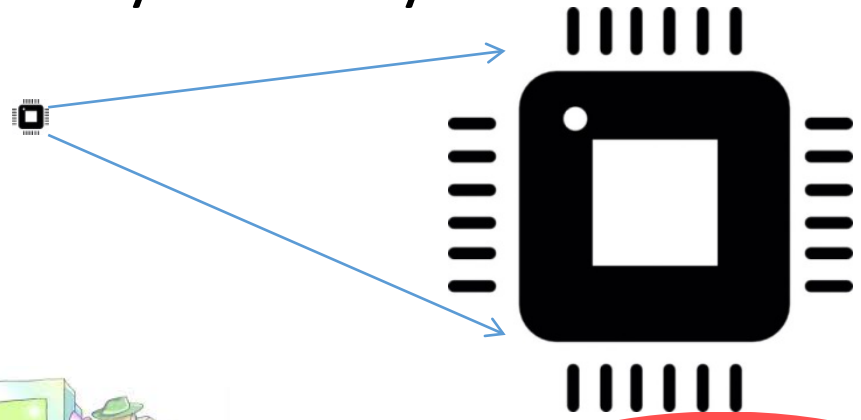
Offline Attacks: A Common Problem

- Password breaches at major companies have affected ~~millions~~ **billions** of user accounts.

Memory Hard Function (MHF)

- **Intuition:** computation costs dominated by memory costs

vs.



sCrypt

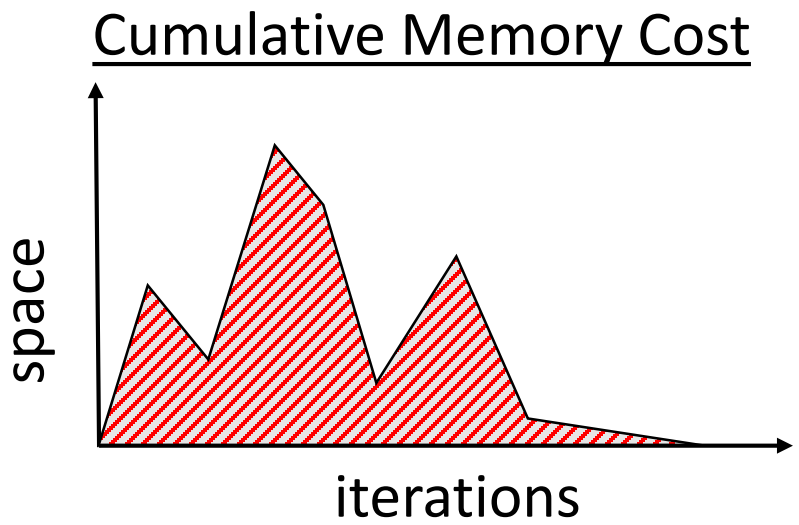
- Data Independent Memory Hard Function (iMHF)
  - Memory access pattern should not depend on input

Measuring Pebbling Costs [AS15]

$$\mathrm{CC}(G) = \min_{\vec{P}} \sum_{i=1}^{t_{\vec{P}}} |P_i|$$

Approximates
Amortized Area x Time
Complexity of iMHF

Memory Used at Step i



Measuring Pebbling Costs [AS15]

- Cumulative Complexity (CC)

Memory Used at Step i

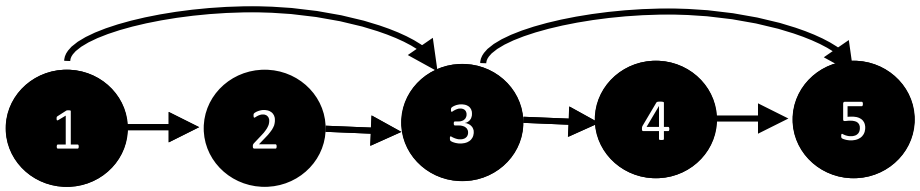
Approximates
Amortized Area x Time
Complexity of iMHF

$$\mathrm{CC}(G) = \min_{\vec{P}} \sum_{i=1}^{t_{\vec{P}}} |P_i|$$

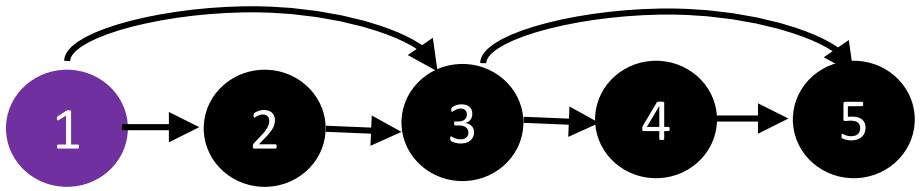
- Guessing two passwords doubles the attackers cost

$$\mathrm{CC}(G, G) = 2 \times \mathrm{CC}(G)$$

Naïve: Pebbling Strategy

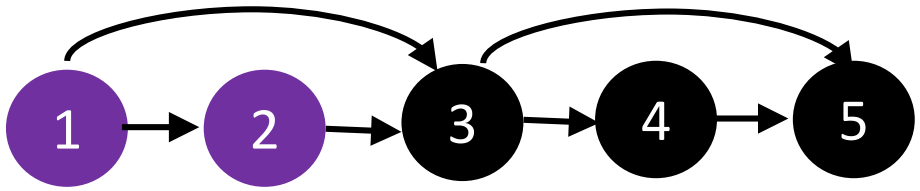


Naïve: Pebbling Strategy



$P_1 = \{1\}$

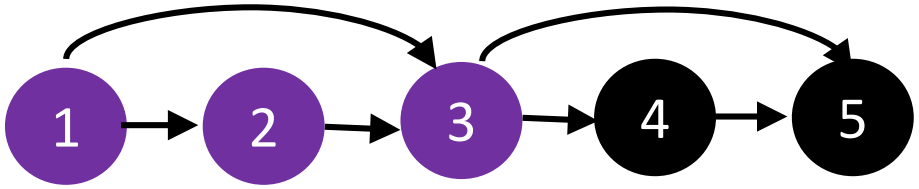
Naïve: Pebbling Strategy



$P_1 = \{1\}$

$P_2 = \{1,2\}$

Naïve: Pebbling Strategy

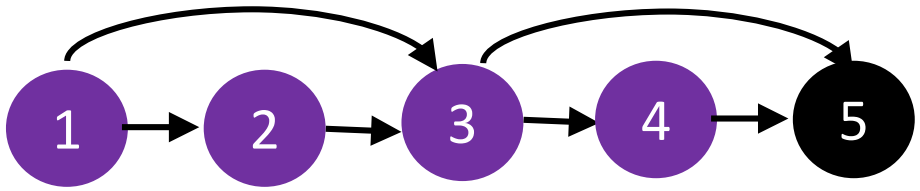


$P_1 = \{1\}$

$P_2 = \{1,2\}$

$P_3 = \{1,2, 3\}$

Naïve: Pebbling Strategy



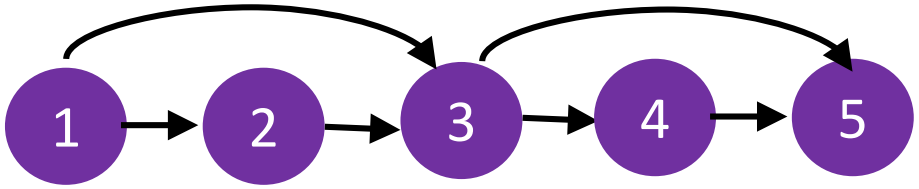
$P_1 = \{1\}$

$P_2 = \{1,2\}$

$P_3 = \{1,2, 3\}$

$P_4 = \{1, 2, 3, 4\}$

Naïve: Pebbling Strategy



$P_1 = \{1\}$

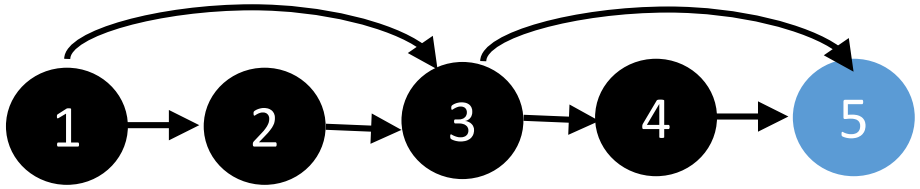
$P_2 = \{1,2\}$

$P_3 = \{1,2, 3\}$

$P_4 = \{1, 2, 3, 4\}$

$P_5 = \{1, 2, 3, 4, 5\}$

Naïve: Pebbling Strategy (CC)



$P_1 = \{1\}$

$P_2 = \{1,2\}$

$P_3 = \{1,2,3\}$

$P_4 = \{1,2,3,4\}$

$P_5 = \{1,2,3,4,5\}$

$$\mathrm{CC}(G) \leq \sum_{i=1}^{5} |P_i|$$

$$= 1 + 2 + 3 + 4 + 5$$

$$= 15$$

Naïve Pebbling Algorithms

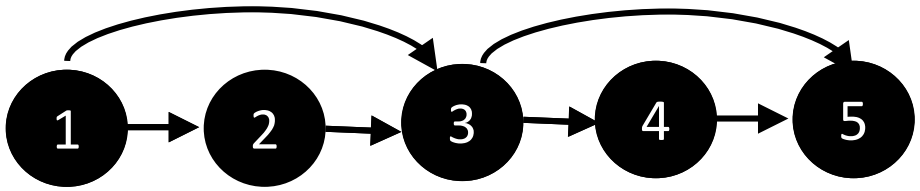
- Naïve (Pebble in Topological Order)
  - Never discard pebbles
  - Legal Pebbling Strategy for any DAG!
  - Pebbling Time: n
  - Sequential: Place one new pebble on the graph in each round

Theorem: *Any DAG G has* $CC(G) \leq \sum_{i \leq n} i = \frac{n(n+1)}{2}$

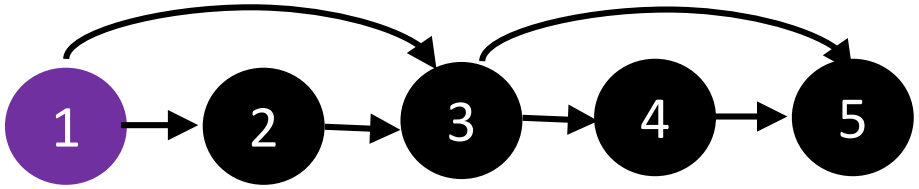
Proof: *Naïve pebbling strategy is legal strategy for any DAG G!*

Question: Can we find a DAG G with $CC(G) = \Omega(n^2)$?

Improved Pebbling

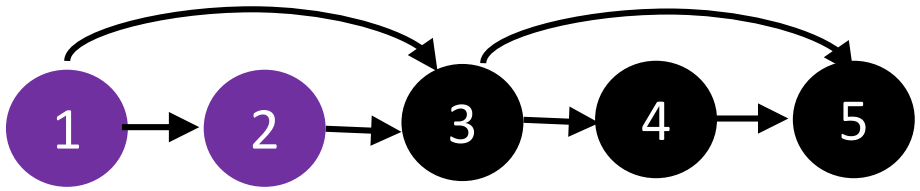


Improved Pebbling



$P_1 = \{1\}$

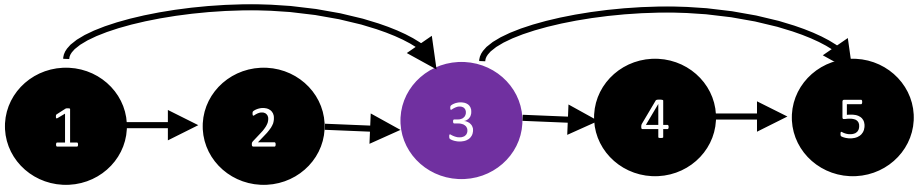
Improved Pebbling



$P_1 = \{1\}$

$P_2 = \{1,2\}$

Improved Pebbling

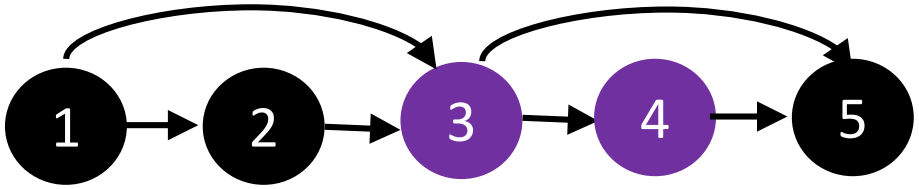


$P_1 = \{1\}$

$P_2 = \{1,2\}$

$P_3 = \{3\}$

Improved Pebbling



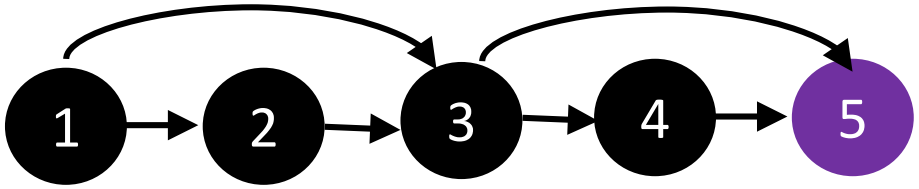
$P_1 = \{1\}$

$P_2 = \{1,2\}$

$P_3 = \{3\}$

$P_4 = \{3,4\}$

Improved Pebbling



$P_1 = \{1\}$

$P_2 = \{1,2\}$

$P_3 = \{3\}$

$P_4 = \{3,4\}$

$P_5 = \{5\}$

Graphs with High CC

Theorem: *Any DAG G has* $CC(G) \leq \sum_{i \leq n} i = \frac{n(n+1)}{2}$

Proof: *Naïve pebbling strategy is legal strategy for any DAG G!*

Question: Can we find a DAG G with $CC(G) = \Omega(n^2)$?

Claim: The complete DAG has $CC(G) \geq \sum_{i \leq n-1} i = \frac{n(n-1)}{2} = \Omega(n^2)$?

Proof: Consider the round immediately before we first place a pebble on node i+1. We must have had pebbles on all of the nodes {1,…,i}.

Question: Can we find a DAG G with $CC(G) = \Omega(n^2)$ and low indegree?

Why do we care about indegree?

In practice the random oracle is instantiated with a function $H: \{0,1\}^{2\lambda} \rightarrow \{0,1\}^{\lambda}$
Label of node v is obtained by hashing labels of v's parents.

Node v has two parents (u and w) ➔ $L_v = H(L_u, L_w)$ ➔ One oracle to H used to compute label

Node v has three parents (u, w, x) ➔ $L_v = H(H(L_u, L_w), L_x)$ ➔ Two oracle queries to H to compute label

Node v has four parents (u, w, x, y) ➔ $L_v = H(H(H(L_u, L_w), L_x), L_y)$ ➔ Three oracle queries to H to compute label

Node v has k parents ➔ k-1 oracle queries to H to compute label

Running time to evaluate $f_{G,H}$ is proportional to $n \times indeg(G)$

Desiderata

Find a DAG G on n nodes such that

1. Constant Indegree ($\delta = 2$)
   - Running Time: $n(\delta - 1) = n$

2. CC(G) $\geq \frac{n^2}{\tau}$ for some small value τ.

Maximize costs for fixed running time n
(Users are impatient)

Outline

- Motivation
- Data Independent Memory Hard Functions (iMHFs)
- **Our Attacks**
  - General Attack on Non Depth Robust DAGs
  - Existing iMHFs are not Depth Robust
  - Ideal iMHFs don’t exist
- Subsequent Results (Depth-Robustness is Sufficient)
- Open Questions

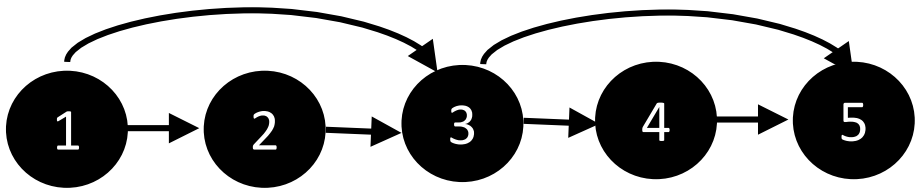
Depth-Robustness: A Necessary Property

Depth Robustness

Definition: A DAG G=(V,E) is (e,d)-reducible if there exists $S \subseteq V$ s.t. $|S| \leq e$ and depth(G-S) $\leq$ d.

Otherwise, we say that G is (e,d)-depth robust.

Example: (1,2)-reducible

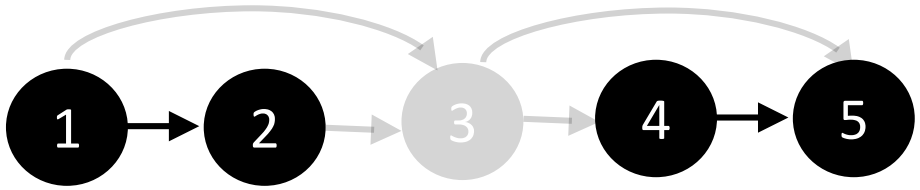


Depth Robustness

Definition: A DAG G=(V,E) is (e,d)-reducible if there exists $S \subseteq V$ s.t. $|S| \leq e$ and depth(G-S) $\leq$ d.

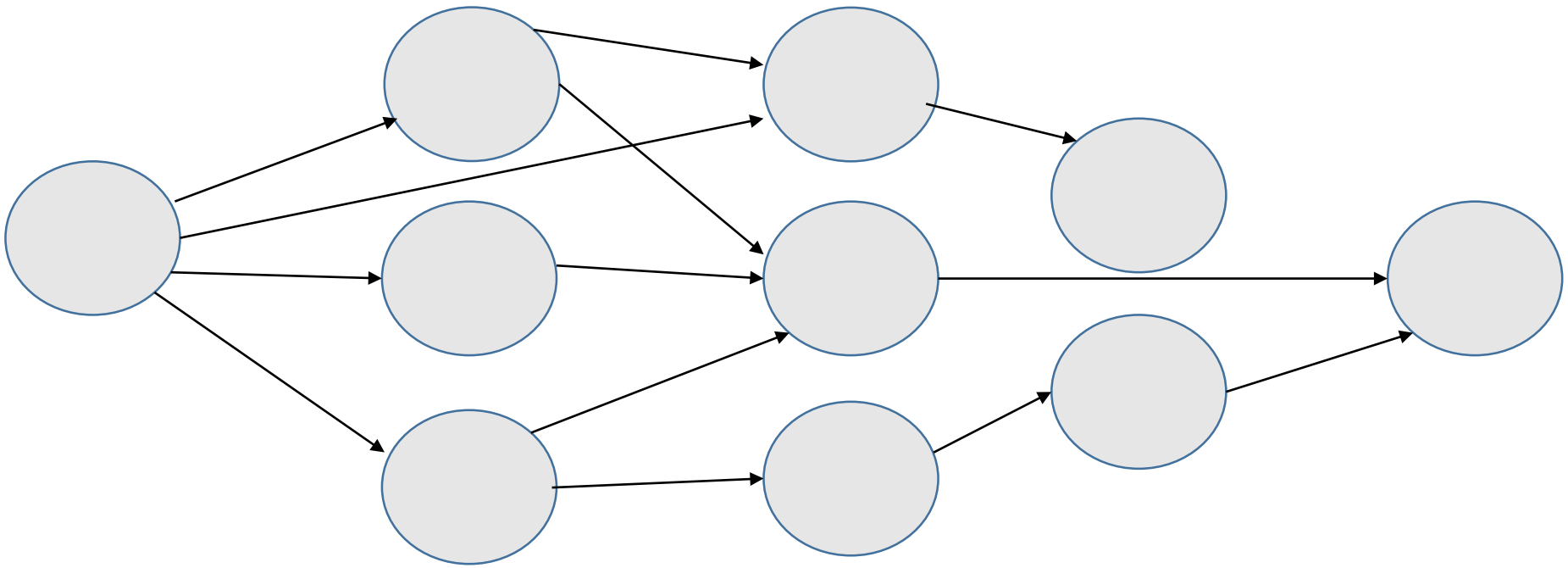
Otherwise, we say that G is (e,d)-depth robust.

Example: (1,2)-reducible



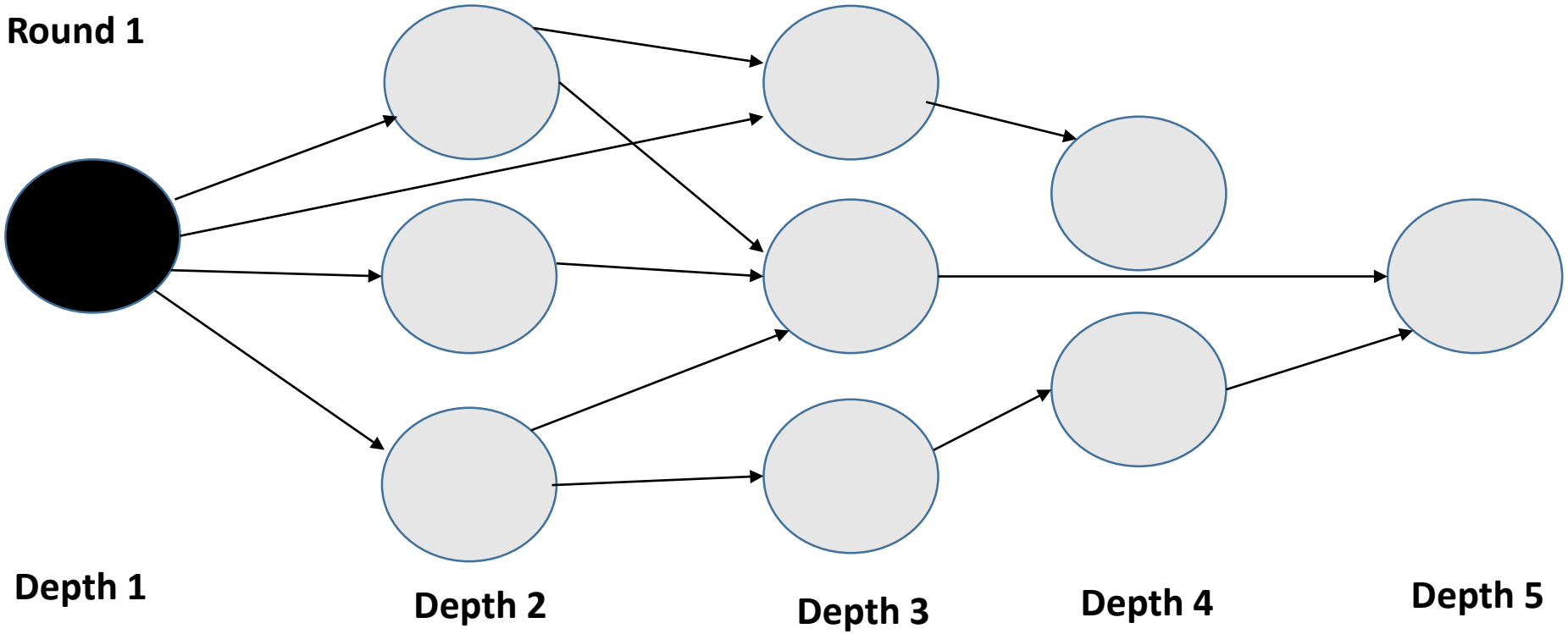
Observation 1

- If a DAG has no directed paths of length d then it can be pebbed in at most d rounds (parallel)



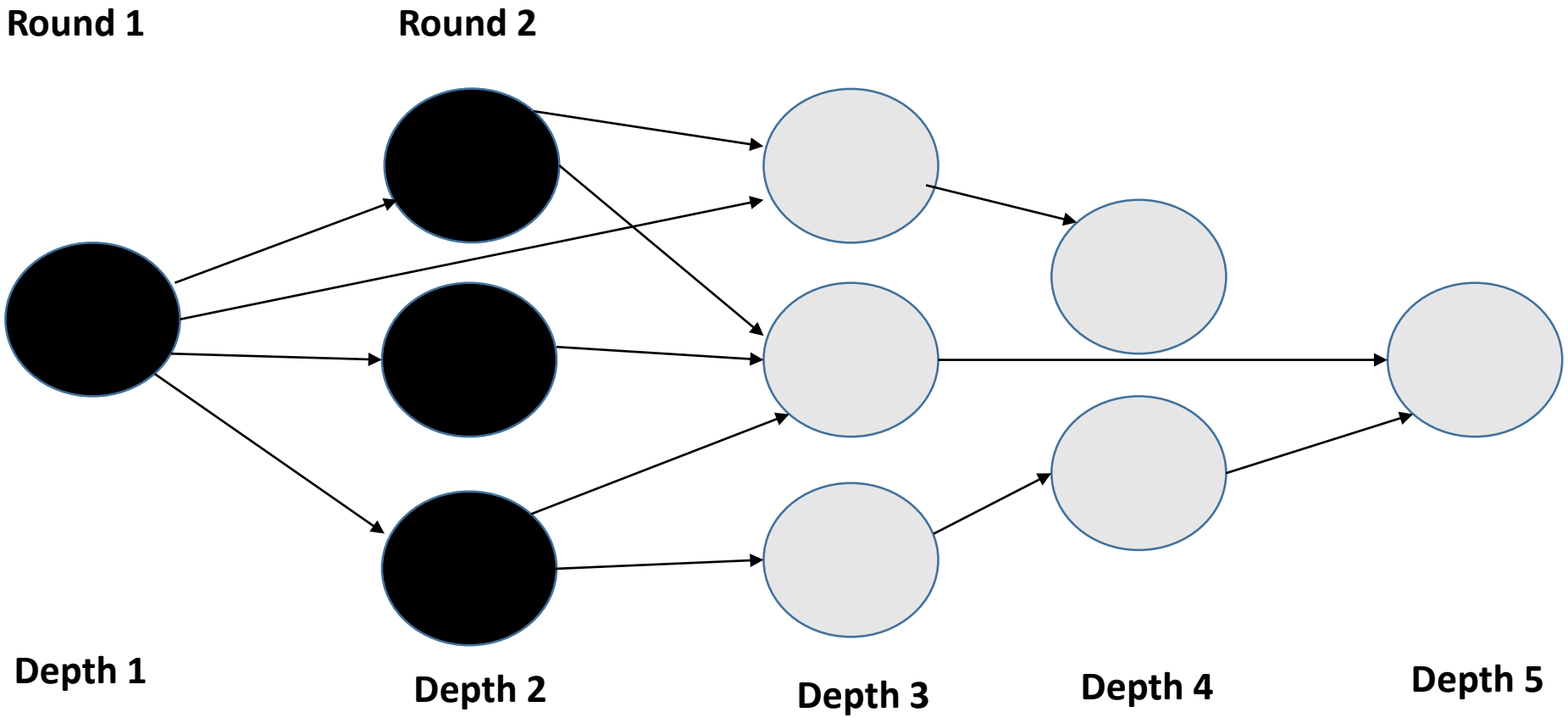
Observation 1

- If a DAG has no directed paths of length d then it can be pebbed in at most d rounds (parallel)



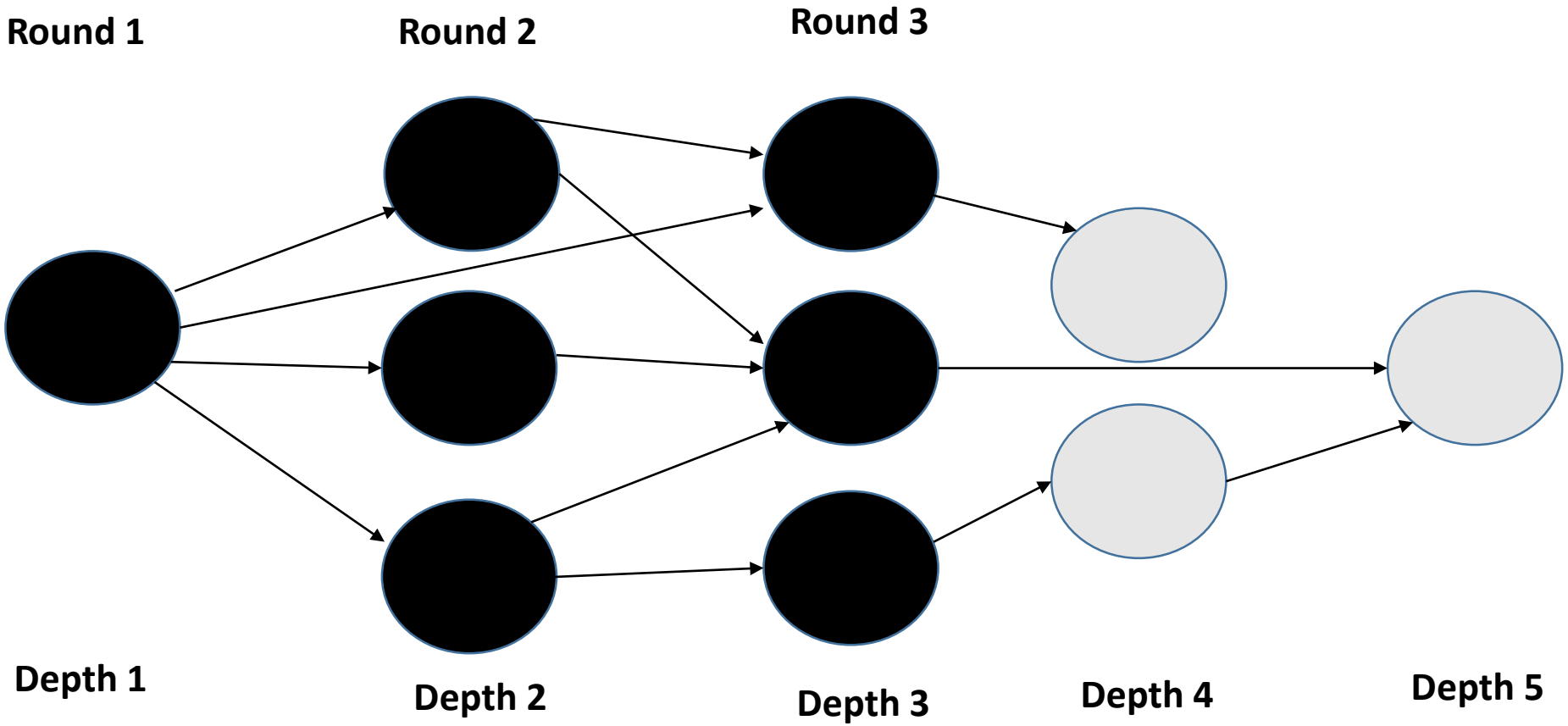
Observation 1

- If a DAG has no directed paths of length d then it can be pebbed in at most d rounds (parallel)



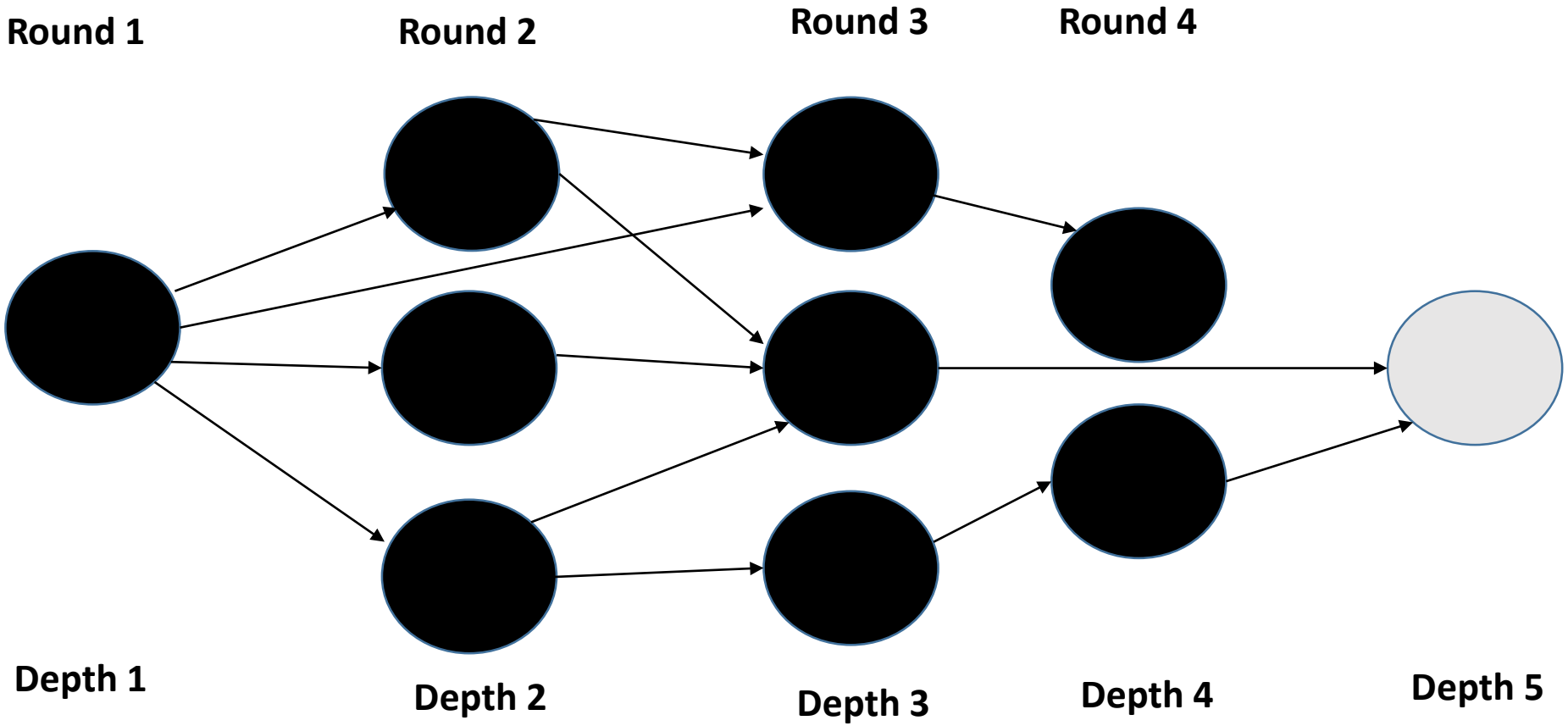
Observation 1

- If a DAG has no directed paths of length d then it can be pebbed in at most d rounds (parallel)



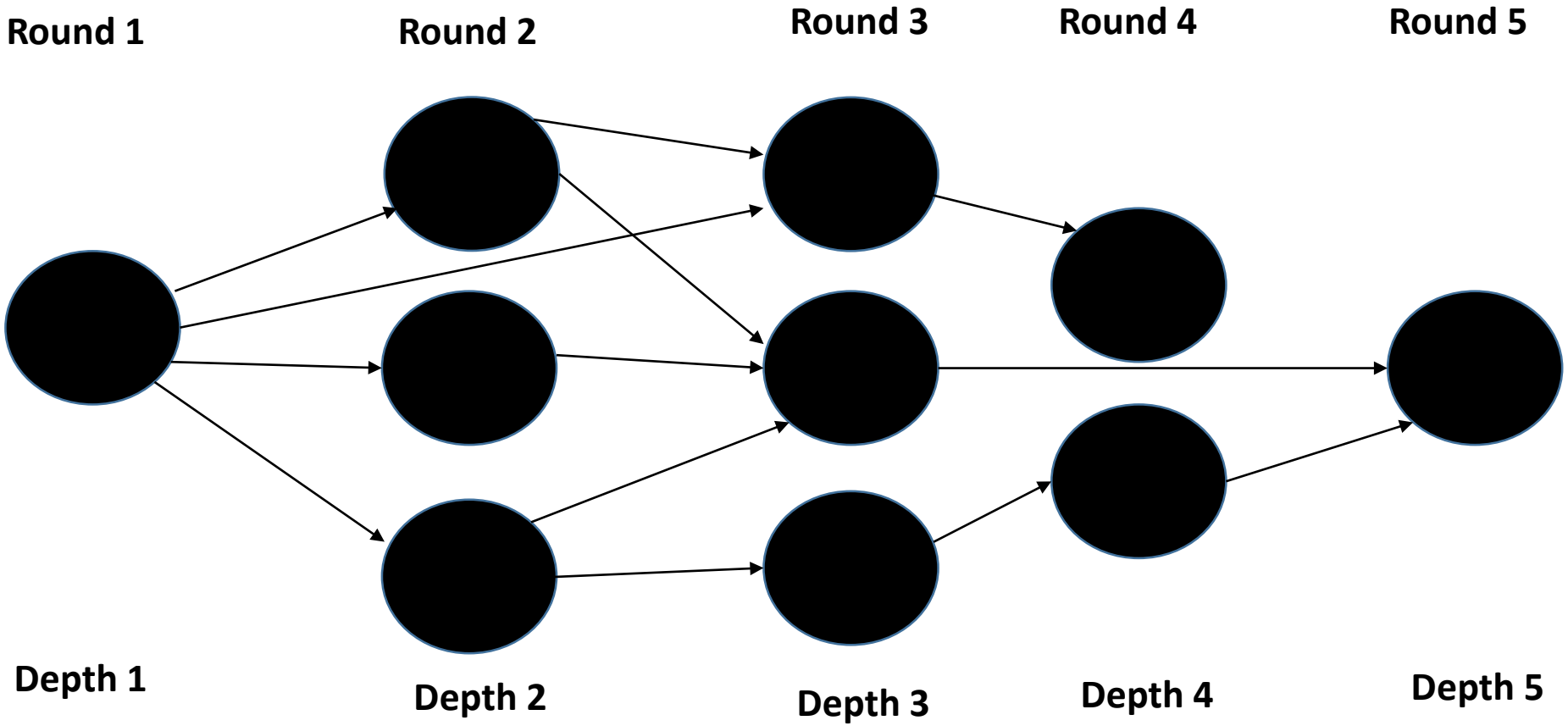
Observation 1

- If a DAG has no directed paths of length d then it can be pebbed in at most d rounds (parallel)



Observation 1

- If a DAG has no directed paths of length d then it can be pebbed in at most d rounds (parallel)



Attacking (e,d)-reducible DAGs

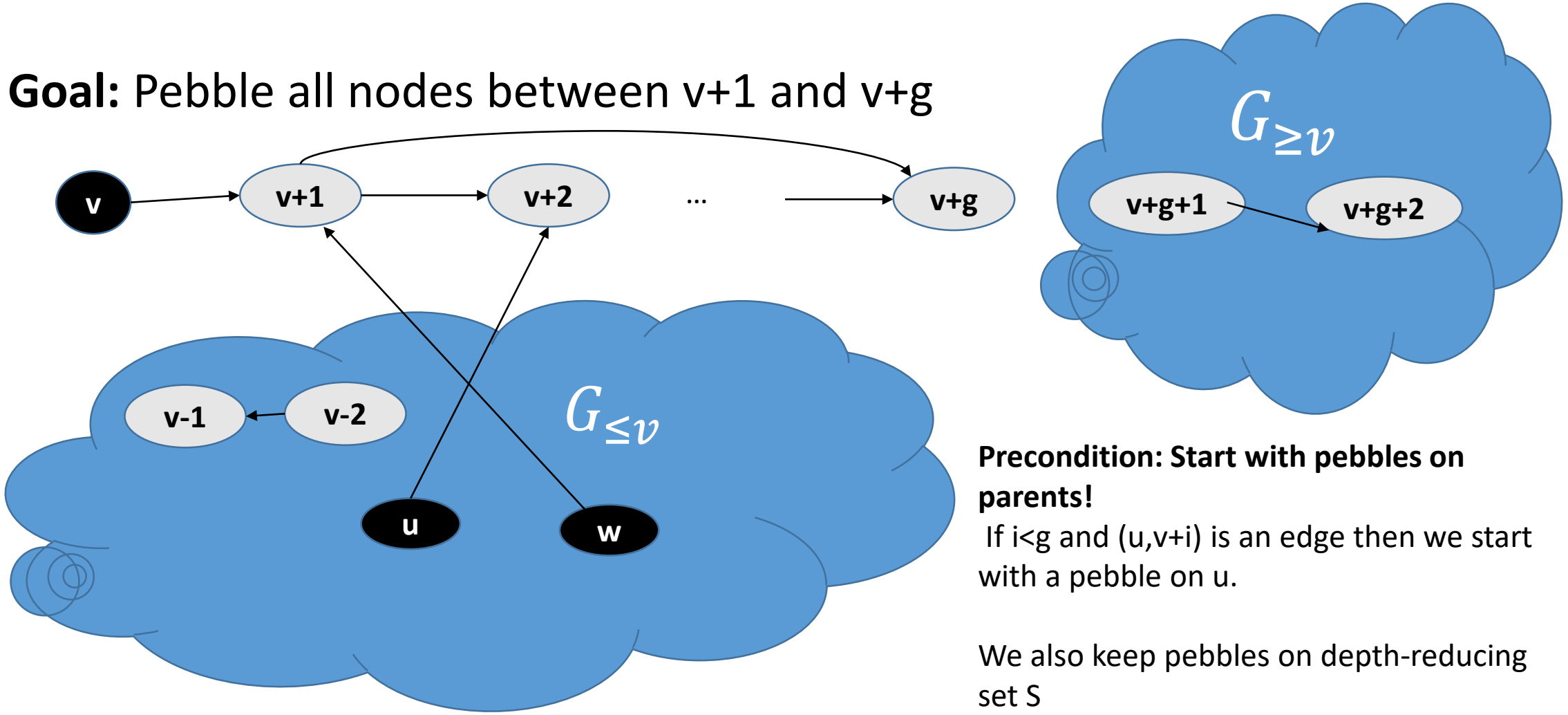
- **Input:** $|S| \leq e$ such that depth(G-S) = d, g > d

- **Light Phase (g rounds)**: Discard most pebbles!
  - Goal: Pebble the next g nodes in g (sequential) steps
  - Low Memory (only keep pebbles on S and on parents of new nodes)
  - Lasts a ``long" time

- **Balloon Phase (d rounds):** Greedily Recover Missing Pebbles
  - Goal: Recover needed pebbles for upcoming light phase
  - Expensive, but quick (at most d steps in parallel).

Light Phase

- **Goal:** Pebble all nodes between v+1 and v+g

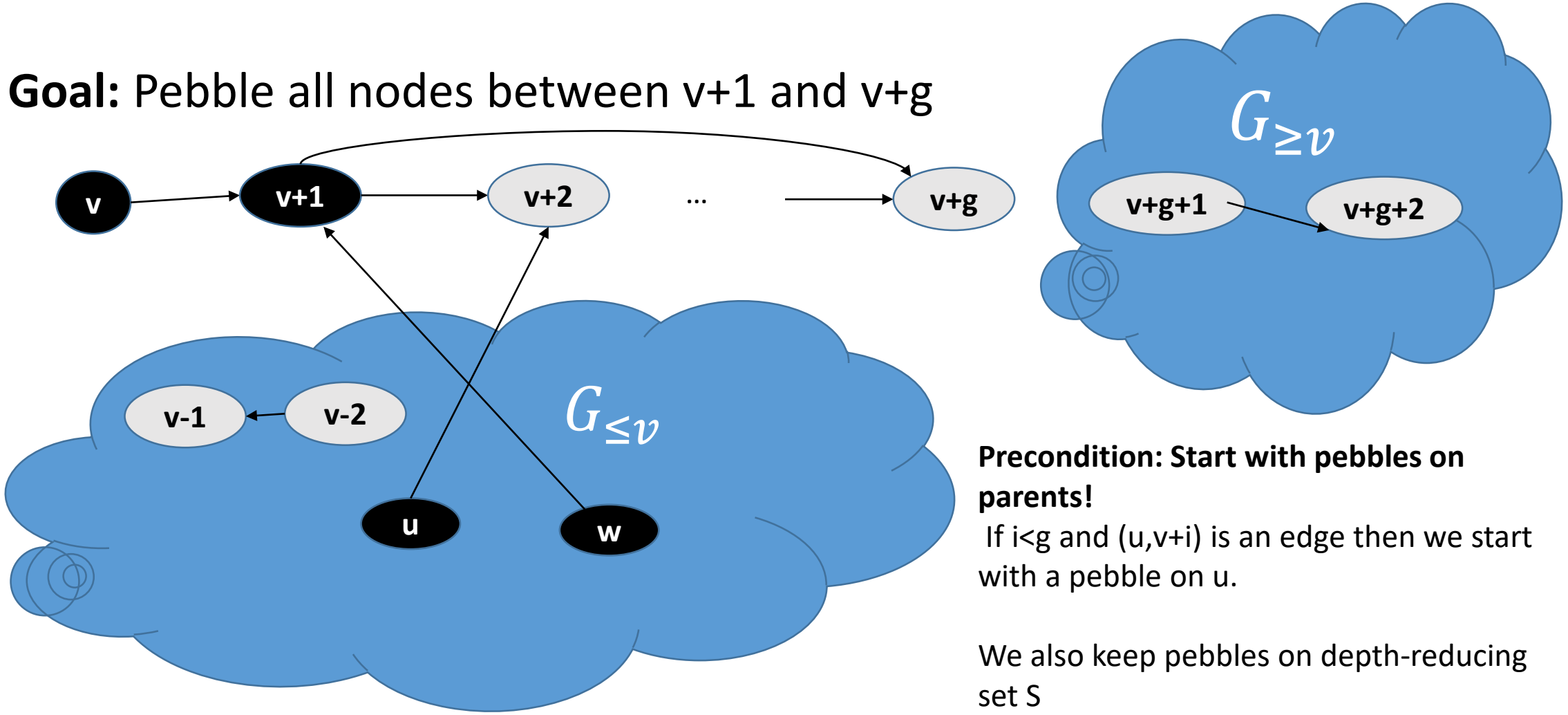


Precondition: Start with pebbles on parents!
If i<g and (u,v+i) is an edge then we start with a pebble on u.

We also keep pebbles on depth-reducing set S

Light Phase

- **Goal:** Pebble all nodes between v+1 and v+g

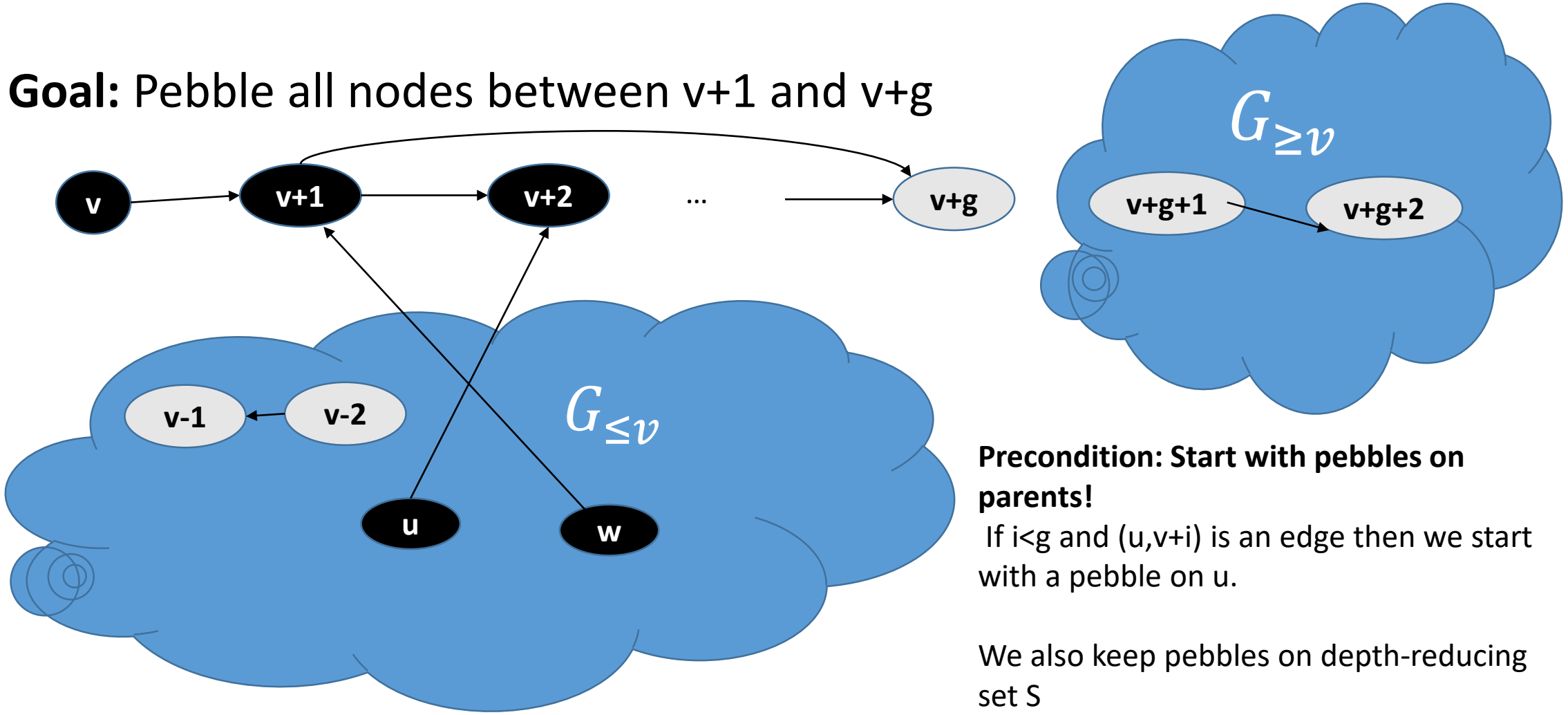


Precondition: Start with pebbles on parents!
If i<g and (u,v+i) is an edge then we start with a pebble on u.

We also keep pebbles on depth-reducing set S

Light Phase

- **Goal:** Pebble all nodes between v+1 and v+g

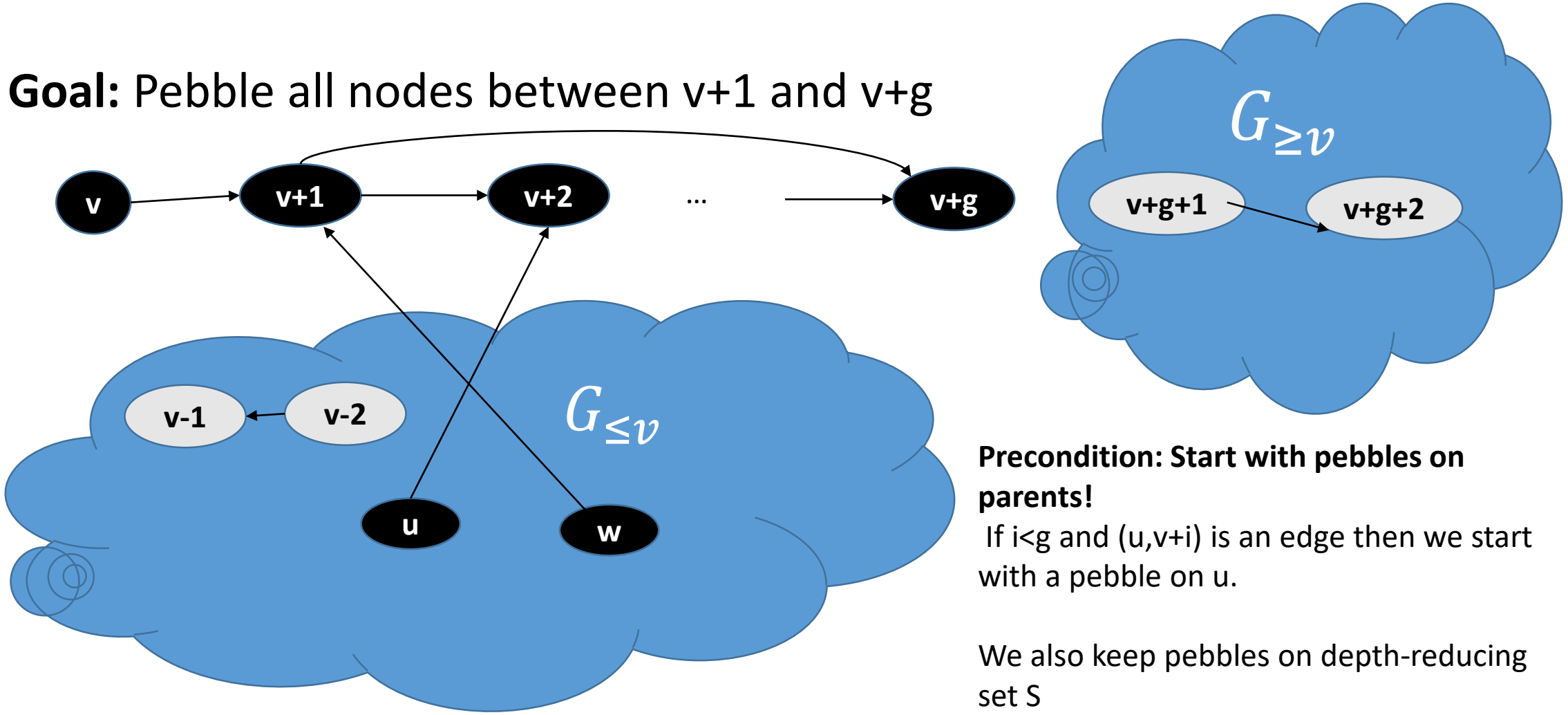


Precondition: Start with pebbles on parents!
If i<g and (u,v+i) is an edge then we start with a pebble on u.

We also keep pebbles on depth-reducing set S

Light Phase

- **Goal:** Pebble all nodes between v+1 and v+g

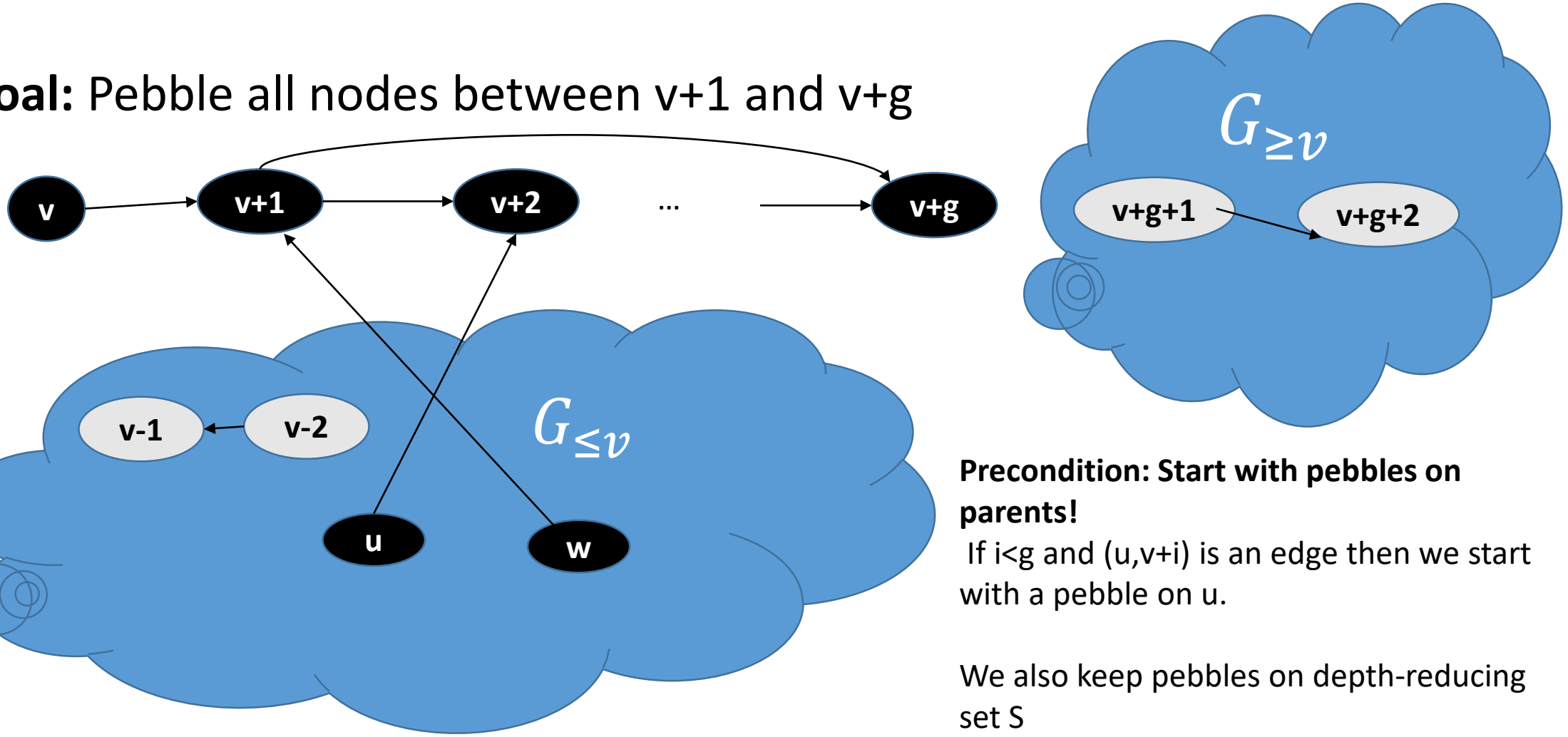


Precondition: Start with pebbles on parents!
If i<g and (u,v+i) is an edge then we start with a pebble on u.

We also keep pebbles on depth-reducing set S

Light Phase

- **Goal:** Pebble all nodes between v+1 and v+g



Precondition: Start with pebbles on parents!
If i<g and (u,v+i) is an edge then we start with a pebble on u.

We also keep pebbles on depth-reducing set S

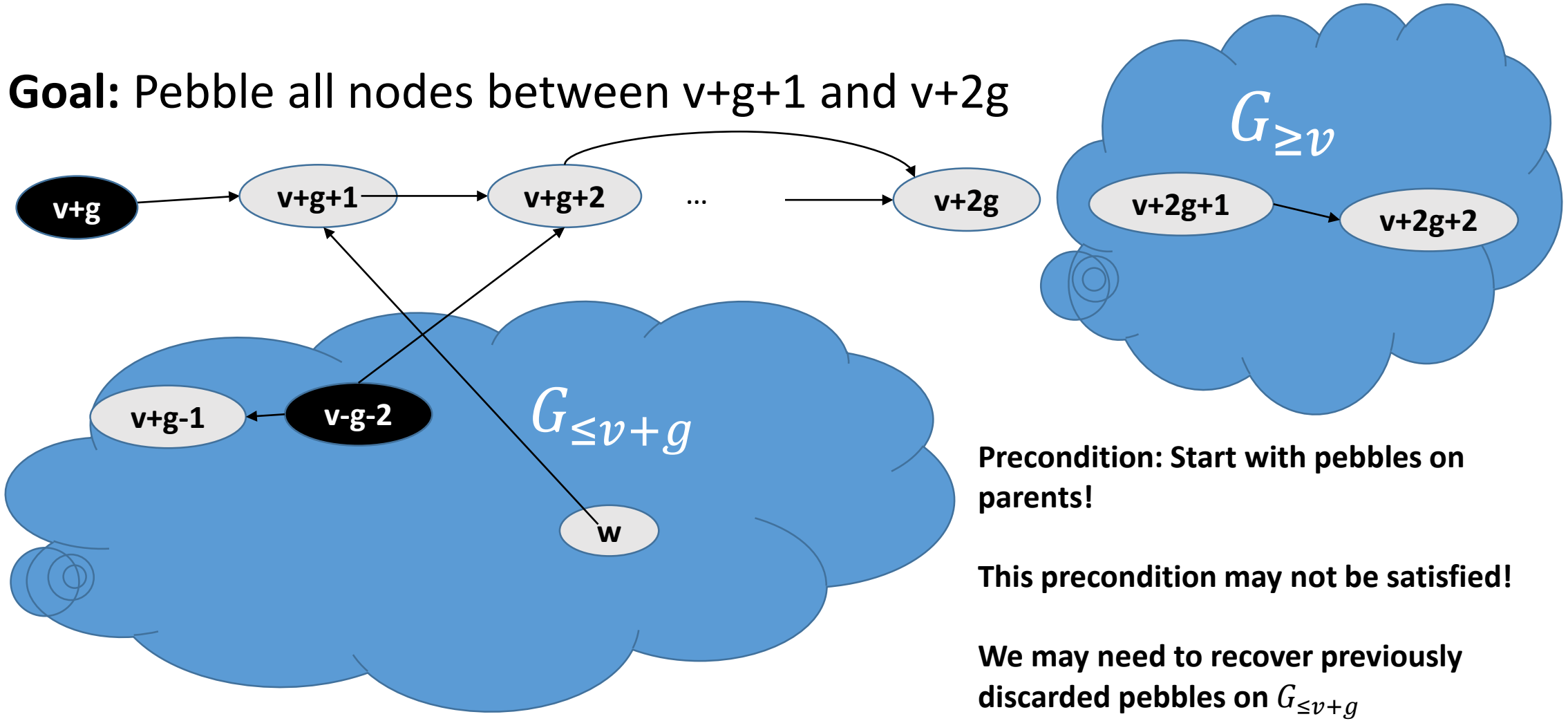
Light Phase Time: g rounds

Max Space Usage: $O(g+e)$

Contribution to CC: $O(g^2+ge)$

Next Light Phase?

- **Goal:** Pebble all nodes between v+g+1 and v+2g



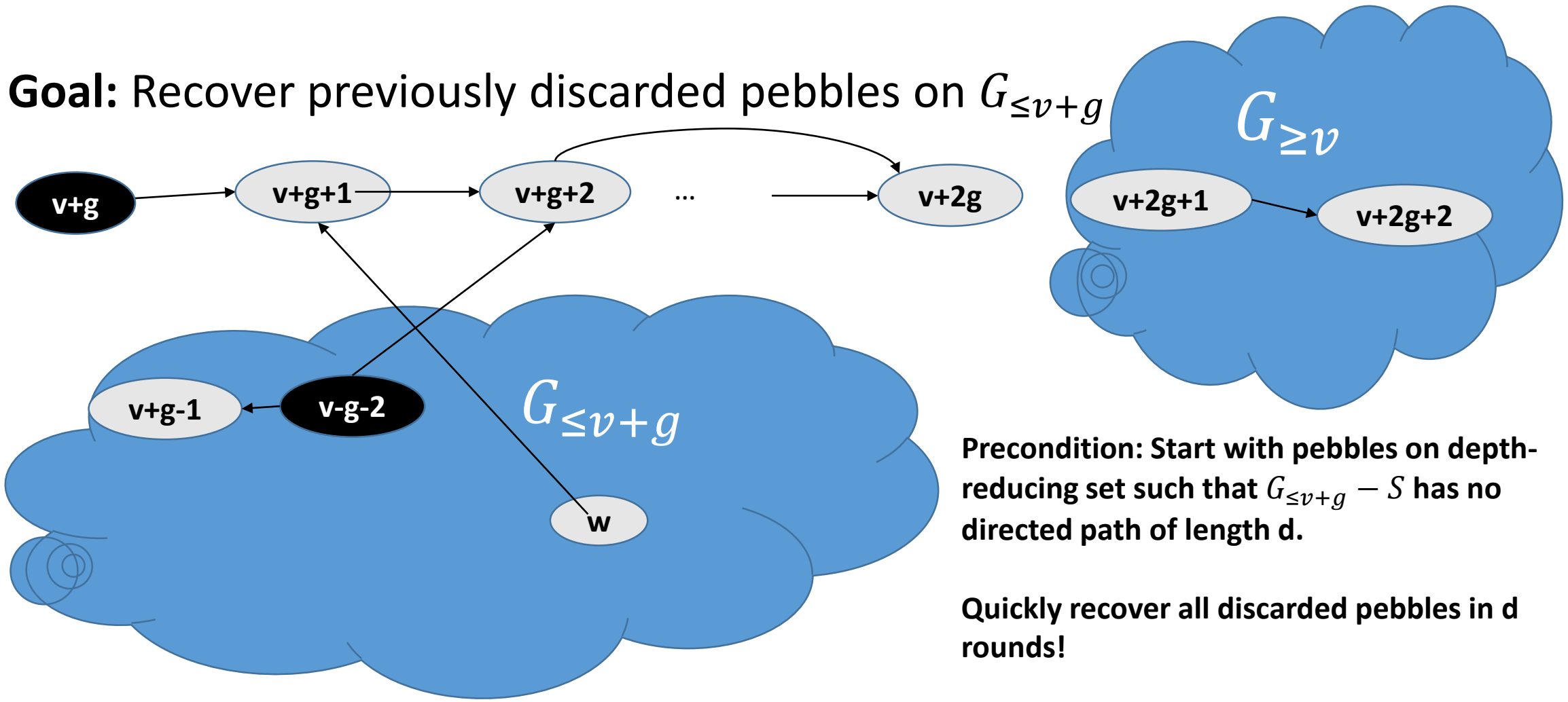
Precondition: Start with pebbles on parents!

This precondition may not be satisfied!

We may need to recover previously discarded pebbles on $G_{\leq v+g}$

Balloon Phase

- **Goal:** Recover previously discarded pebbles on $G_{\leq v+g}$



Precondition: Start with pebbles on depth-reducing set such that $G_{\leq v+g} - S$ has no directed path of length d.

Quickly recover all discarded pebbles in d rounds!

Balloon Phase Time: d rounds **Max Space Usage:** $v + g \leq N$ **Contribution to CC:** $O(dN)$

Attacking (e,d)-reducible DAGs

Algorithm 1: GenPeb (G, S, g, d)

Arguments: $G = (V, E)$, $S \subseteq V$, $g \in [\mathsf{depth}(G - S), n]$, $d \geq \mathsf{depth}(G - S)$

1 **for** $i = 1$ *to* n **do**
2 Pebble node i.
3 $l \leftarrow \lfloor i/g \rfloor * g + d + 1$
4 **if** $i \mod g \in [d]$ **then** // Balloon Phase
5 $d' \leftarrow d - (i \mod g) + 1$
6 $N \leftarrow \mathsf{need}(l, l + g, d')$
7 Pebble every $v \in N$ which has all parents pebbled.
8 Remove pebble from any $v \notin K$ where $K \leftarrow S \cup \mathsf{keep}(i, i + g) \cup \{n\}$.
9 **else** // Light Phase
10 $K \leftarrow S \cup \mathsf{parents}(i, i + g) \cup \{n\}$
11 Remove pebbles from all $v \notin K$.
12 **end**
13 **end**

Main Theorem

Theorem (Depth-Robustness is a necessary condition): If G is (e,d)-reducible then is an (efficient) attack A such that

$$\mathrm{E_R}(A) \leq en + \delta gn + \frac{n}{g}nd + n\mathrm{R} + \frac{n}{g}n\mathrm{R}.$$

Main Theorem

Theorem (Depth-Robustness is a necessary condition): If G is (e,d)-reducible then is an (efficient) attack A such that

$$\mathrm{E_R}(A) \leq en + \delta gn + \frac{n}{g}nd + n\mathrm{R} + \frac{n}{g}n\mathrm{R}.$$

Upper bounds pebbles on nodes x $\in S$, where

$|S| = e$

depth(G-S) $\leq d$

#pebbling rounds

Main Theorem

Theorem (Depth-Robustness is a necessary condition): If G is (e,d)-reducible then is an (efficient) attack A such that

$$\mathrm{E}_{\mathrm{R}}(A) \leq en + \delta gn + \frac{n}{g} nd + n\mathrm{R} + \frac{n}{g} n\mathrm{R}.$$

Maintain pebbles on parents of next g nodes to be pebbled. Each node has at most δ incoming edges

#pebbling rounds

Main Theorem

Theorem (Depth-Robustness is a necessary condition): If G is not (e,d)-node robust then is an (efficient) attack A such that

$$\mathrm{E}_{\mathrm{R}}(A) \leq en + \delta gn + \frac{n}{g} nd + n\mathrm{R} + \frac{n}{g} n\mathrm{R}.$$

#balloon phases

Max #pebbles on G
In each round of balloon phase

Length of a balloon phase

Main Theorem

Theorem (Depth-Robustness is a necessary condition): If G is not (e,d)-node robust then is an (efficient) attack A such that

$$\mathrm{E_R}(A) \leq en + \delta gn + \frac{n}{g}nd + n\mathrm{R} + \frac{n}{g}n\mathrm{R}.$$

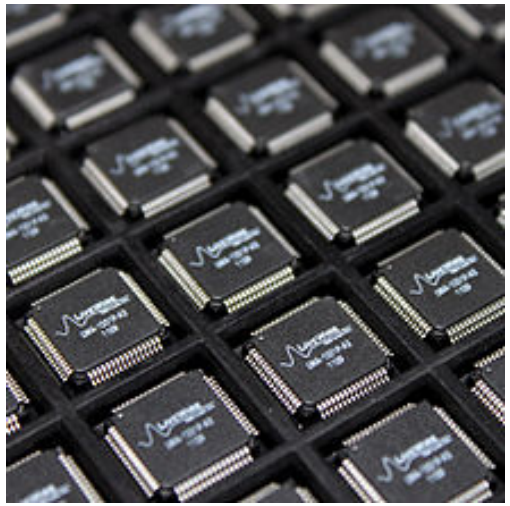
Set $g = \sqrt{nd}$

$$\mathrm{E_R}(A) = \mathrm{O}\left(en + \sqrt{n^3 d}\right).$$

In particular, $\mathrm{E_R}(A) = \mathrm{o}(n^2)$ for e,d=o(n).

Question

Are existing iMHF candidates based on depth-robust DAGs?



iMHF Candidates

- Catena [FLW15]
  - Special Recognition at Password Hashing Competition
  - Two Variants: Dragonfly and Double-Butterfly
  - Security proofs in sequential space-time model

- Balloon Hashing [CBS16]
  - Newer proposal (three variants in original proposal)

- Argon2 [BDK15]
  - Winner of the Password Hashing Competition
  - Argon2i (data-independent mode) is recommended for Password Hashing

- This Talk: Focus on Argon2i-A (version from Password Hashing Competition)
  - Attack ideas do extend to Argon2i-B (latest version)

Attack Outline

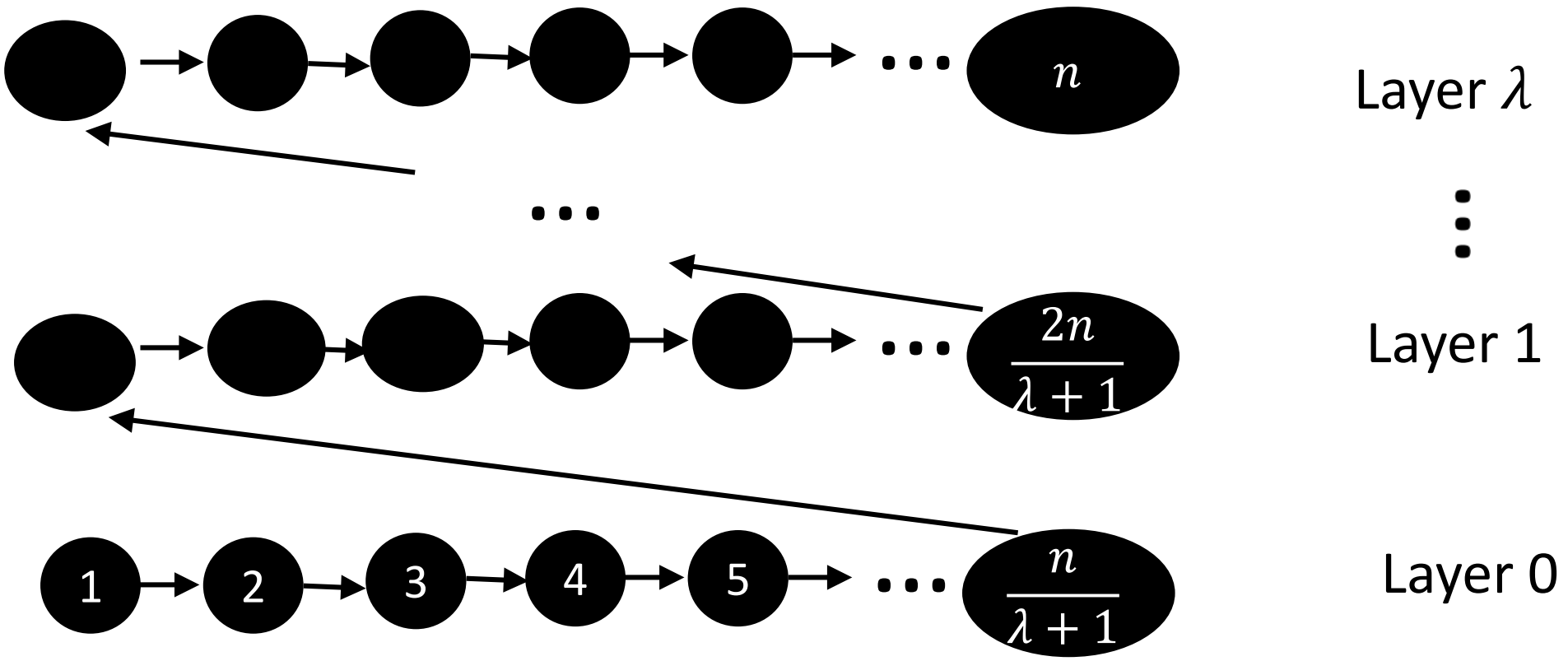
- Show that any “layered DAG” is reducible
  - Note: Catena DAGs are layered DAGs

- Show that an Argon2i DAG is *almost* a “layered DAG.”
  - Turn Argon2i into layered DAG by deleting a few nodes
  - Hence, an Argon2i DAG is also reducible.

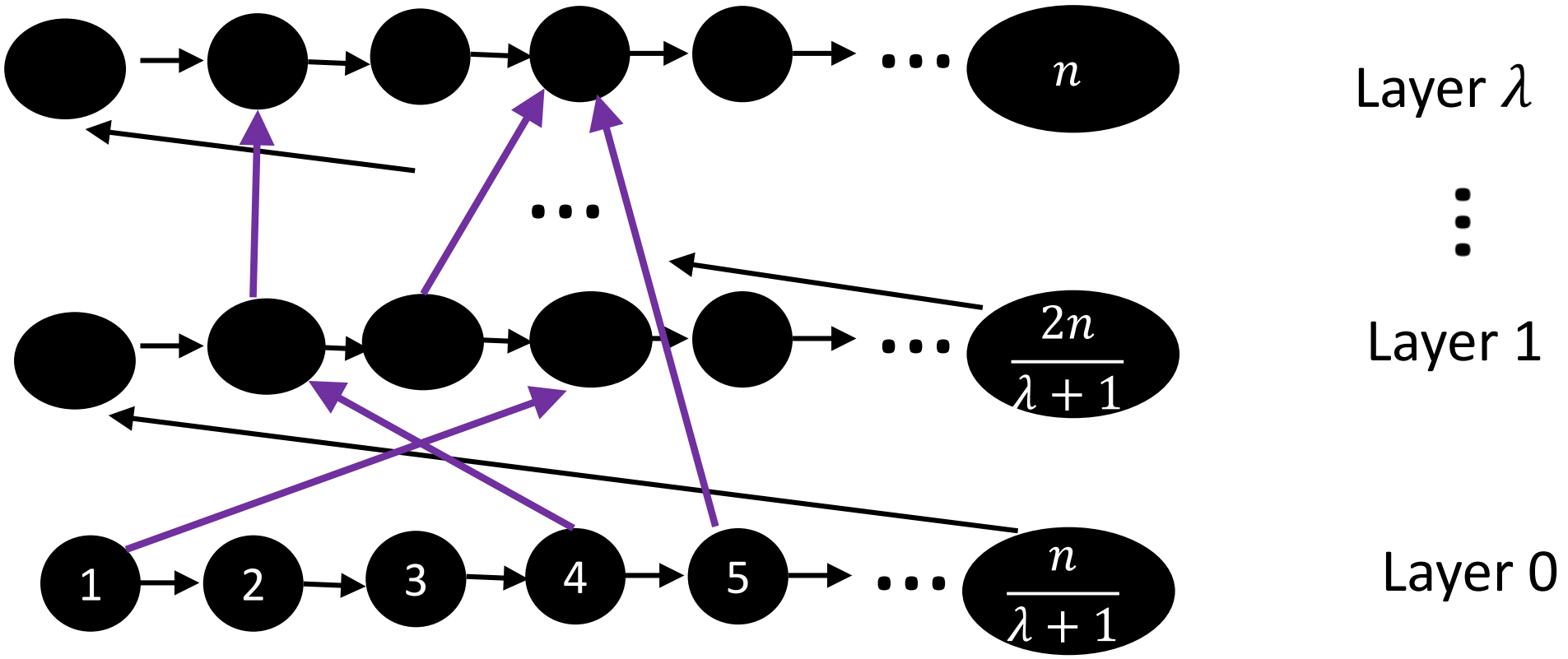
Catena

- Catena Bit Reversal DAG ($\mathrm{BRG}_{\lambda}^{n}$)
  - λ-layers of nodes ($\lambda \leq 5$)
  - Edges between layers correspond to the bit-reversal operation
  - **Theorem[LT82]:** sST(BRG_{1}^{n})$= \Omega(n^2)$
- Catena Butterfly ($\mathrm{DBG}_{\lambda}^{n}$)
  - $\lambda = O(\log n)$-layers of nodes
  - Edges between layers correspond to FFT
  - $\mathrm{DBG}_{\lambda}^{n}$ is a “super-concentrator.”
  - **Theorem[LT82] =>** sST(BRG_{1}^{n})$= \Omega\left(\frac{n^2}{\log(n)}\right)$

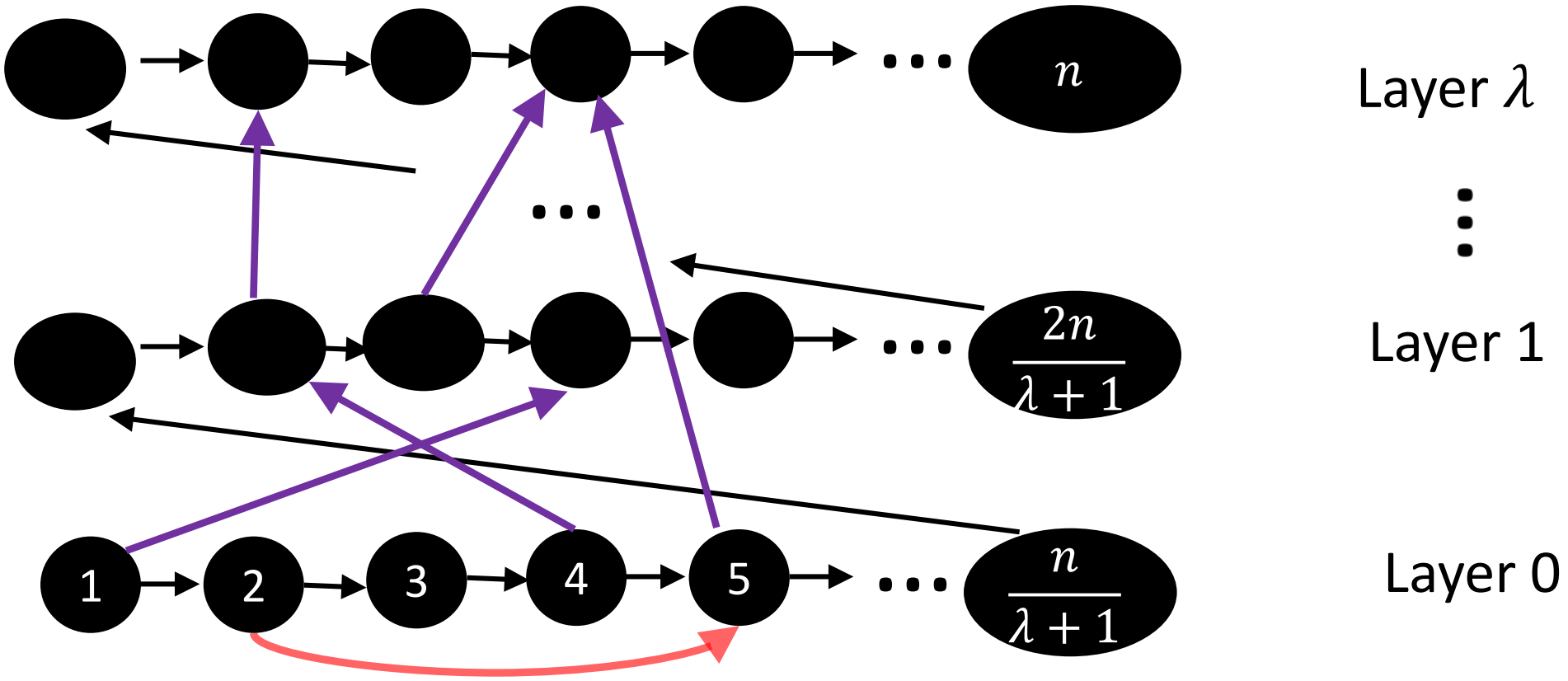
λ-Layered DAG (Catena)



λ-Layered DAG (Catena)



λ-Layered DAG (Catena)

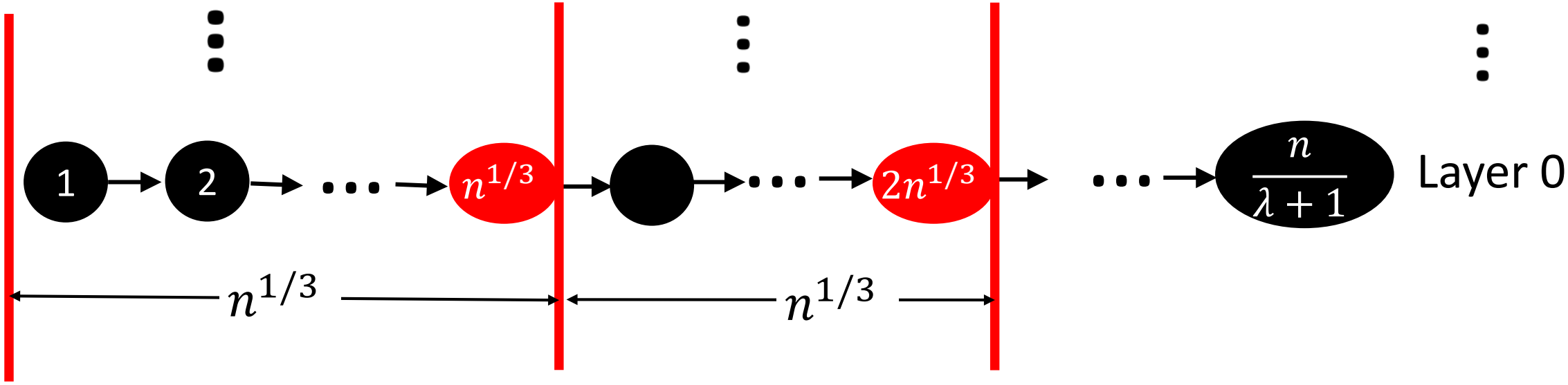


Disallowed! All edges must go to a higher layer (except for (i,i+1))

Layered Graphs are Reducible

Theorem (Layered Graphs Not Depth Robust): Let G be a λ-Layered DAG then G is $\left(n^{2/3}, n^{1/3}(\lambda+1)\right)$-reducible.

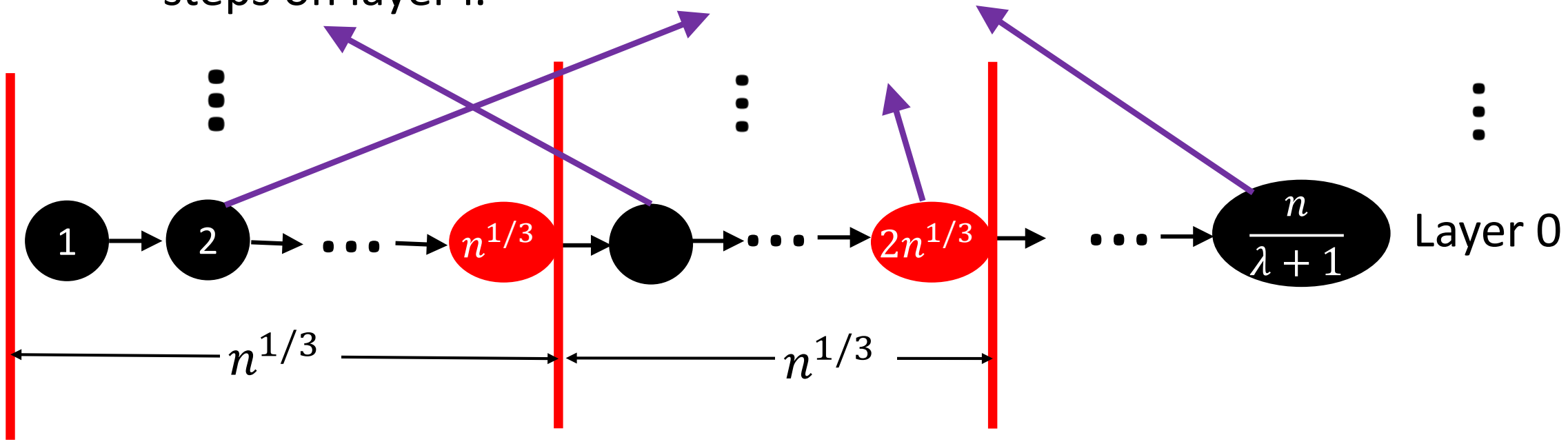
Proof: Let $\boldsymbol{S = \left\{ i \times n^{1/3} \middle| i \leq n^{2/3} \right\}}$ any path p can spend at most $n^{1/3}$ steps on layer i.



Layered Graphs are Reducible

Theorem (Layered Graphs Not Depth Robust): Let G be a λ-Layered DAG then G is $\left(n^{2/3}, n^{1/3}(\lambda+1)\right)$-reducible.

Proof: Let $\boldsymbol{S = \left\{ i \times n^{1/3} \middle| i \leq n^{2/3} \right\}}$ any path p can spend at most $n^{1/3}$ steps on layer i.



Layered Graphs are Reducible

Theorem (Layered Graphs Not Depth Robust): Let G be a λ-Layered DAG then G is $\left(n^{2/3}, n^{1/3}(\lambda+1)\right)$-reducible.

Corollary: $\mathrm{E}_{\mathrm{R}}(G) \leq O\left(\lambda n^{5/3}\right)$.

Attack Quality: $\mathrm{Quality}_{\mathrm{R}}(A) = \Omega\left(\frac{n^{1/3}}{\lambda}\right)$.

Previous Attacks on Catena

- [AS15] $\mathrm{CC}(\mathrm{BRG}_1^n) \leq O(n^{1.5})$
  - Gap between cumulative cost $O(n^{1.5})$ and sequential space-time cost $\Omega(n^2)$
- [BK15] $\mathrm{ST}(\mathrm{BRG}_\lambda^n) \leq O(n^{1.8})$ for $\lambda > 1$.
- Our result $\mathrm{CC}(\mathrm{BRG}_\lambda^n) \leq O(n^{1.67})$ *

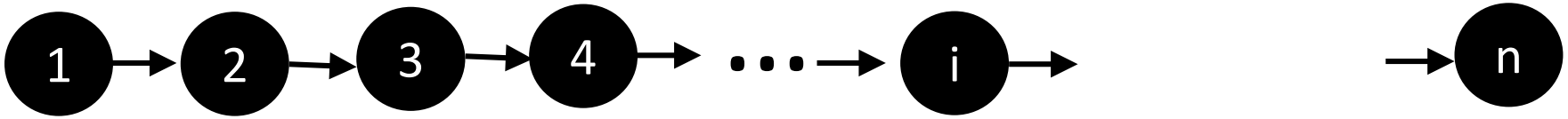
* Applies to all Catena variants.

Argon2i [BDK]

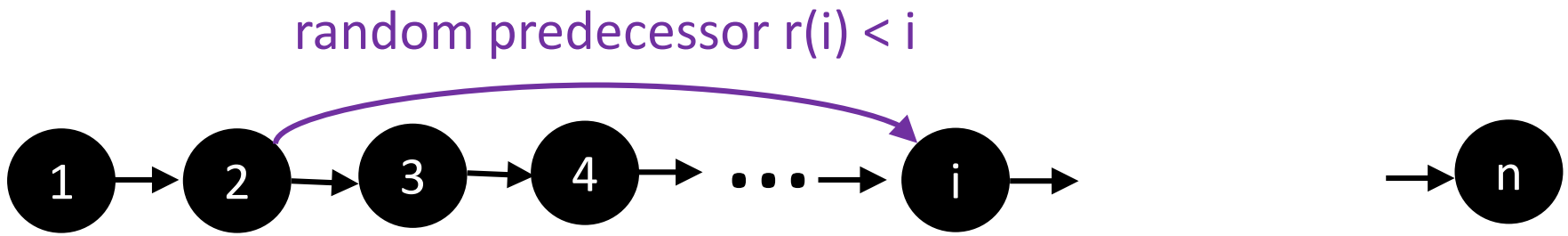
- Argon2: Winner of the password hashing competition[2015]

- Authors recommend Argon2i variant (data-independent) for password hashing.

Argon2i

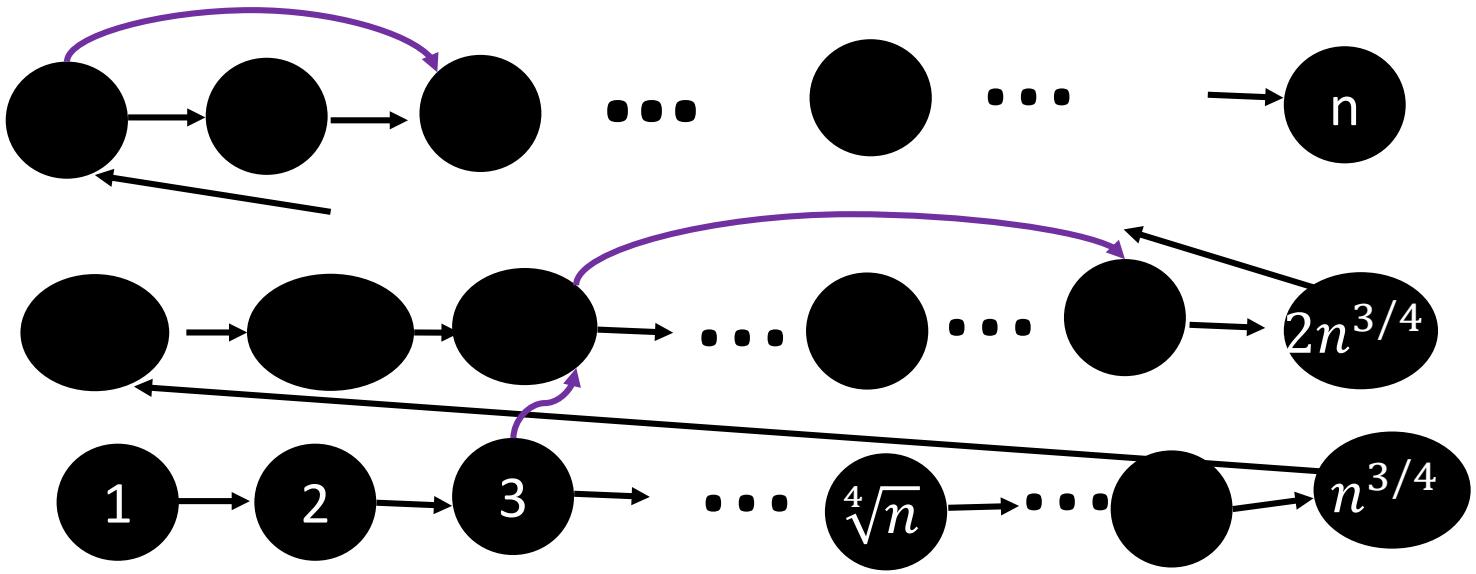


Argon2i



Indegree: $\delta = 2$

Argon2i is a layered DAG (almost)



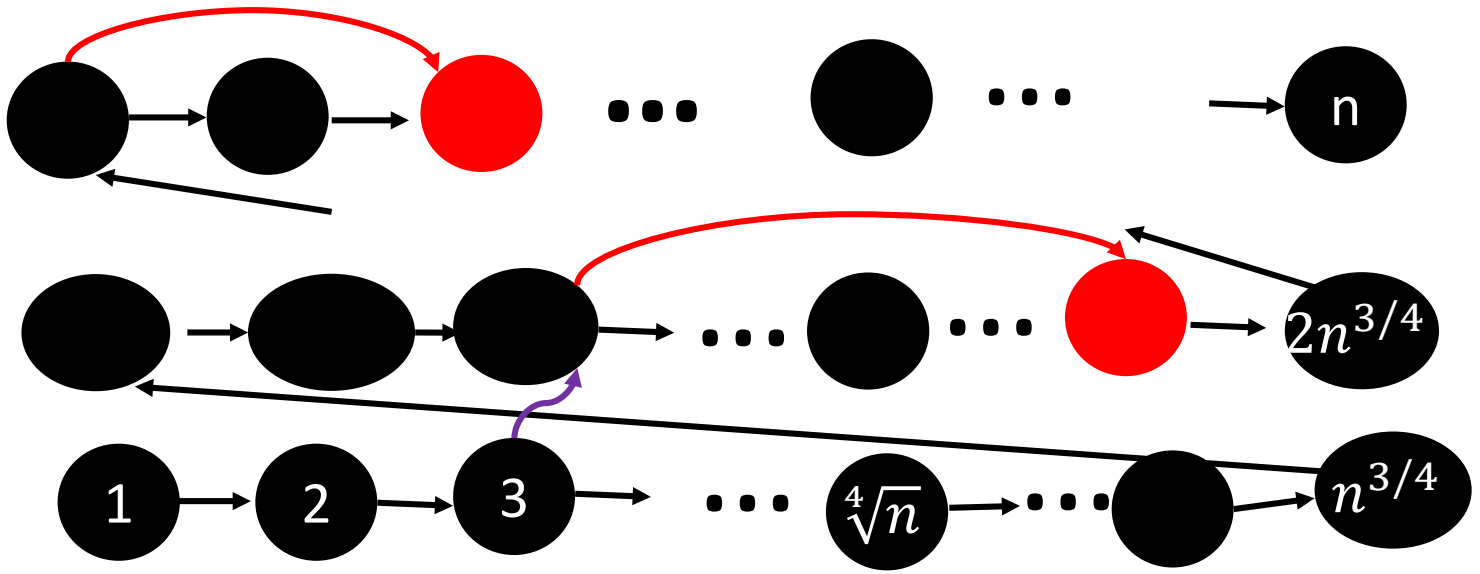
Layer $\sqrt[4]{n}$

Layer 1

Layer 0

Argon2i is a layered DAG (almost)

Definition: $S_2 = \{v_i | \ v_{r(i)} \text{and } v_i \text{ in same layer}\}$



Layer $\sqrt[4]{n}$

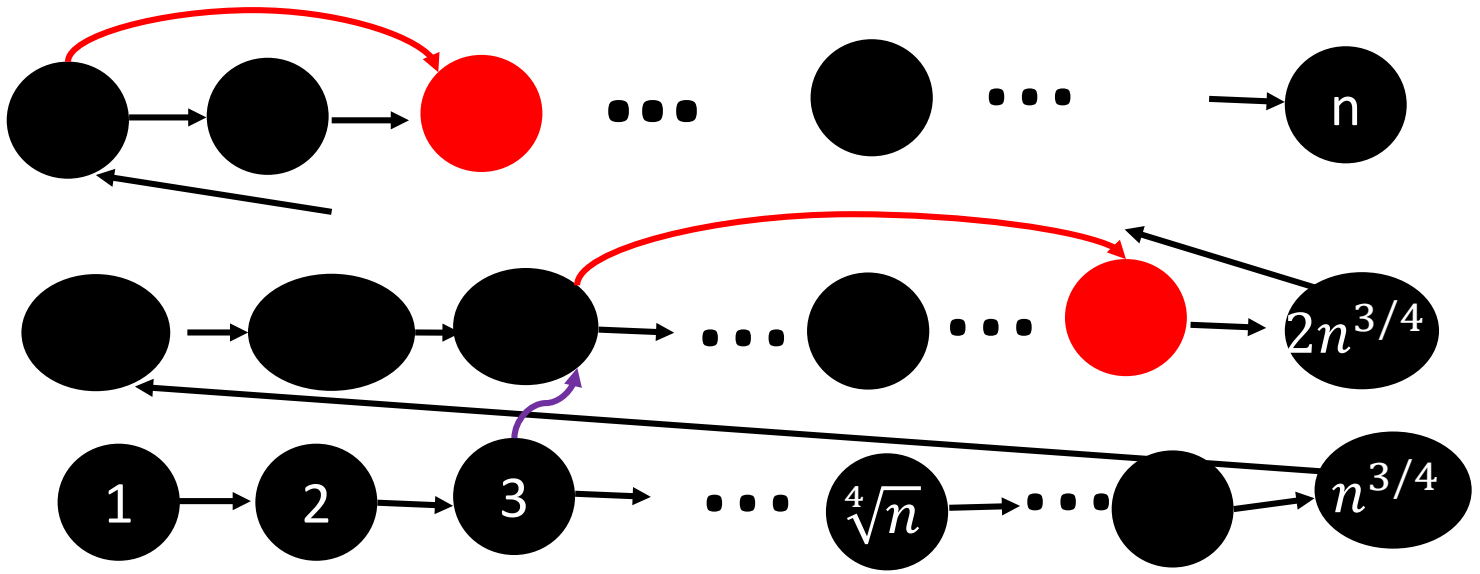
Layer 1

Layer 0

Claim: $\mathrm{E}[S_2] = O(n^{3/4} \log n)$

Argon2i is a layered DAG (almost)

Definition: $S_2 = \{v_i | \ v_{r(i)} \text{and } v_i \text{ in same layer}\}$



Layer $\sqrt[4]{n}$

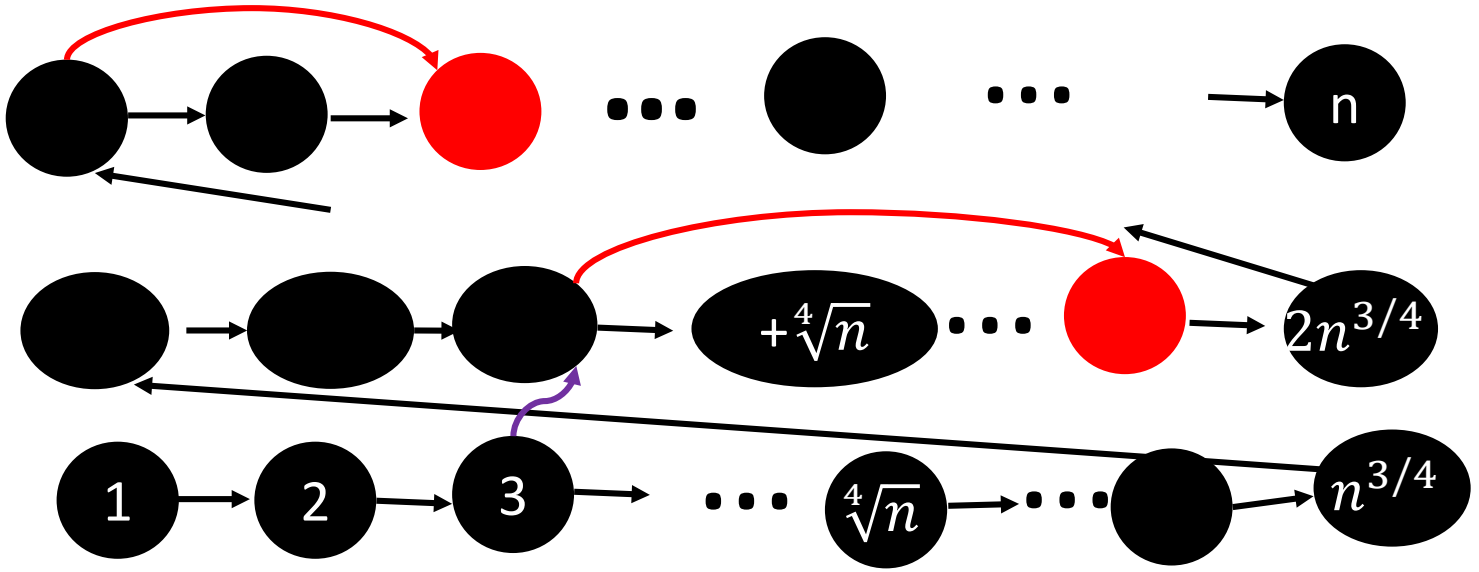
Layer 1

Layer 0

$$\Pr[v \in S_2 | v \text{ in Layer } i] \leq \frac{1}{i} \qquad E[\text{Layer } i \cap S_2] \leq \frac{n^{3/4}}{i}$$

Argon2i is a layered DAG (almost)

Definition: $S_2 = \{v_i | \ v_{r(i)} \text{and } v_i \text{ in same layer}\}$



Layer $\sqrt[4]{n}$

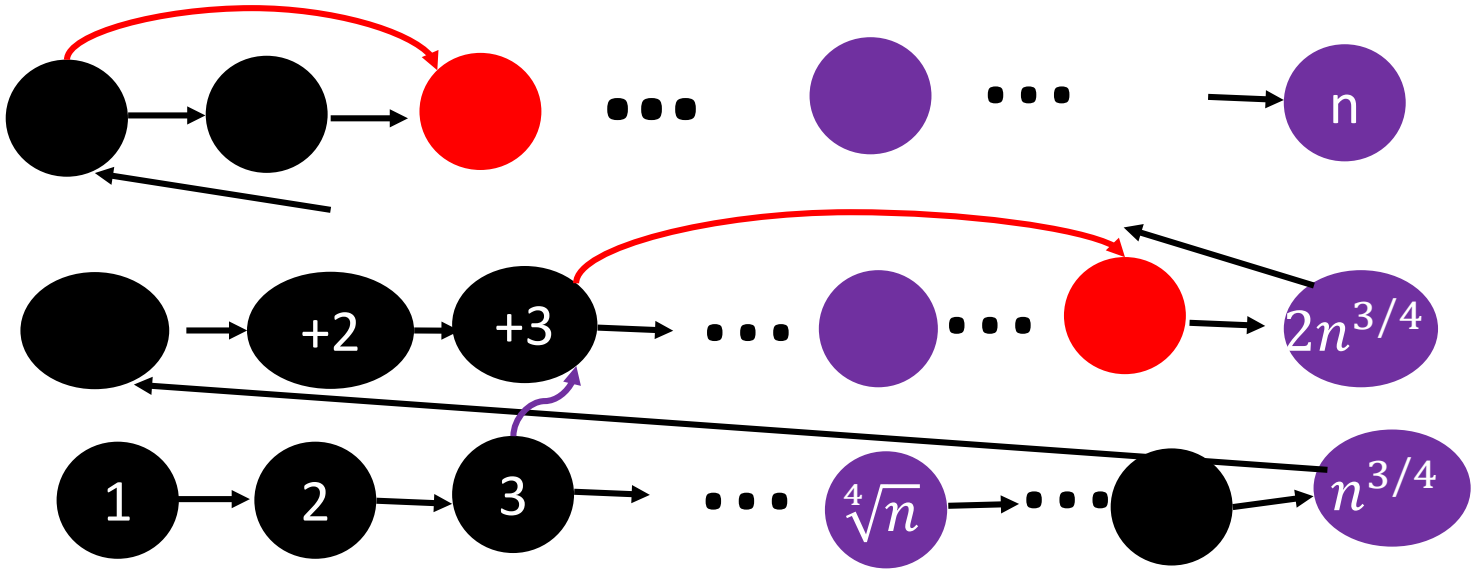
Layer 1

Layer 0

Claim: $\mathrm{E}[S_2] = O(n^{3/4} \log n)$

Argon2i is a layered DAG (almost)

Let S = S_1+S_2



Layer $\sqrt[4]{n}$

⋮

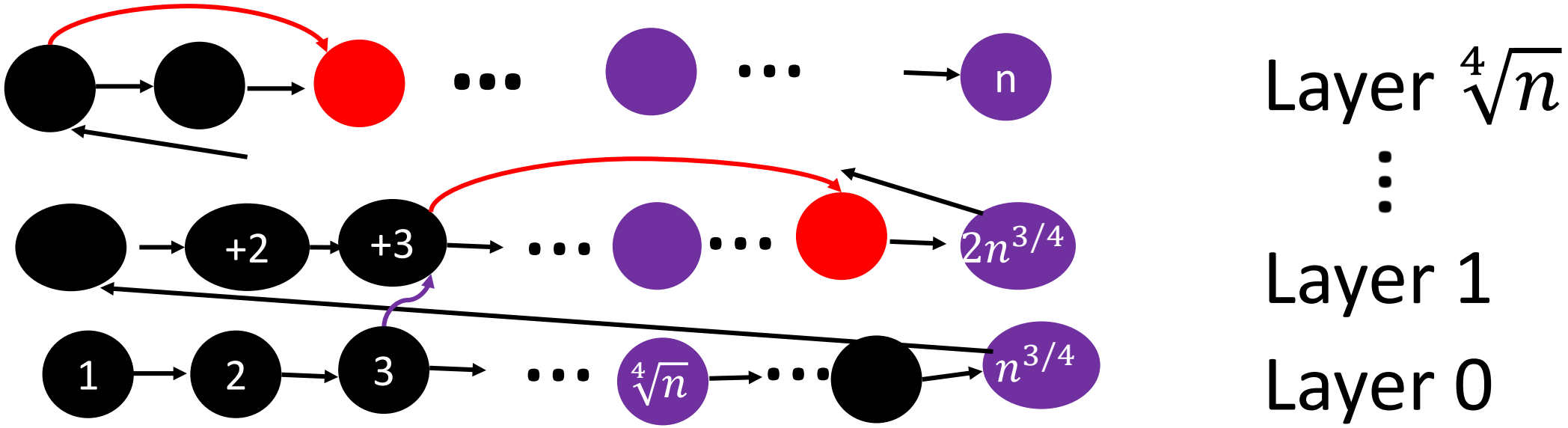
Layer 1

Layer 0

Fact: $\mathrm{E}[S] = O\left(n^{3/4} \log n\right)$ and depth(G-S)$\leq \sqrt{n}$.

Argon2i is a layered DAG (almost)

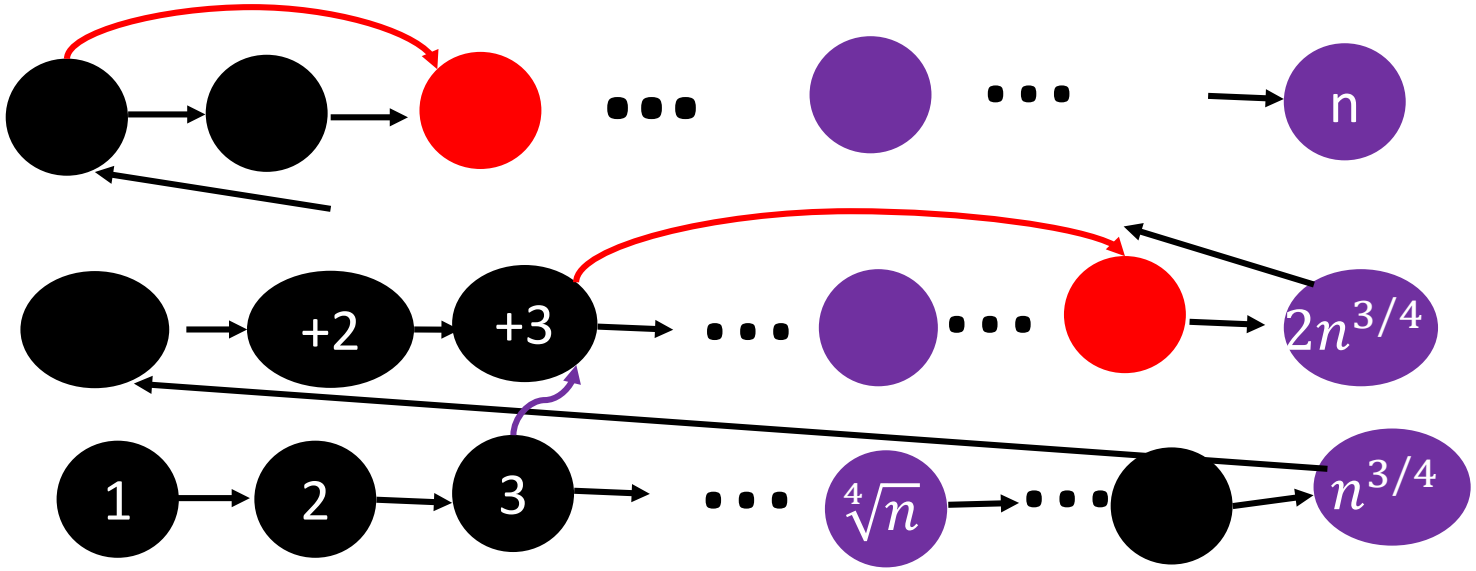
Let S = S_1+S_2



Theorem: G is $(2n^{3/4}\log n, \sqrt{n})$-reducible with high probability.

Argon2i is a layered DAG (almost)

Let S = S_1+S_2



Layer $\sqrt[4]{n}$

⋮

Layer 1

Layer 0

Corollary: $\mathrm{ER}(G) \leq O\left(n^{7/4} \log n\right)$.

$$\mathrm{Quality}_{\mathrm{R}}(A) \leq \Omega\left(\frac{n^{1/4}}{\log n}\right).$$

Ideal iMHFs Don't Exist

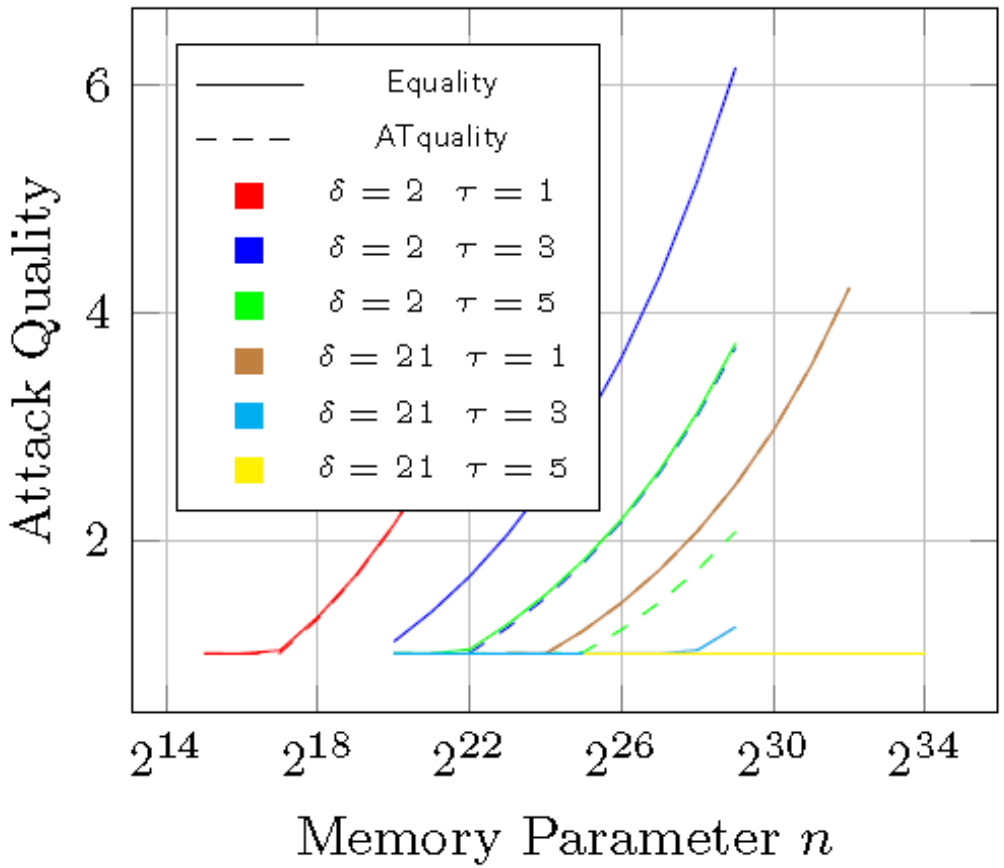
- **Thm:** If G has n nodes and constant in-degree δ=O(1) then G is :

$$\left(O\left(\frac{n \log\log n}{\log(n)}\right),\ \frac{n}{\log^2 n}\right)\text{-reducible.}$$

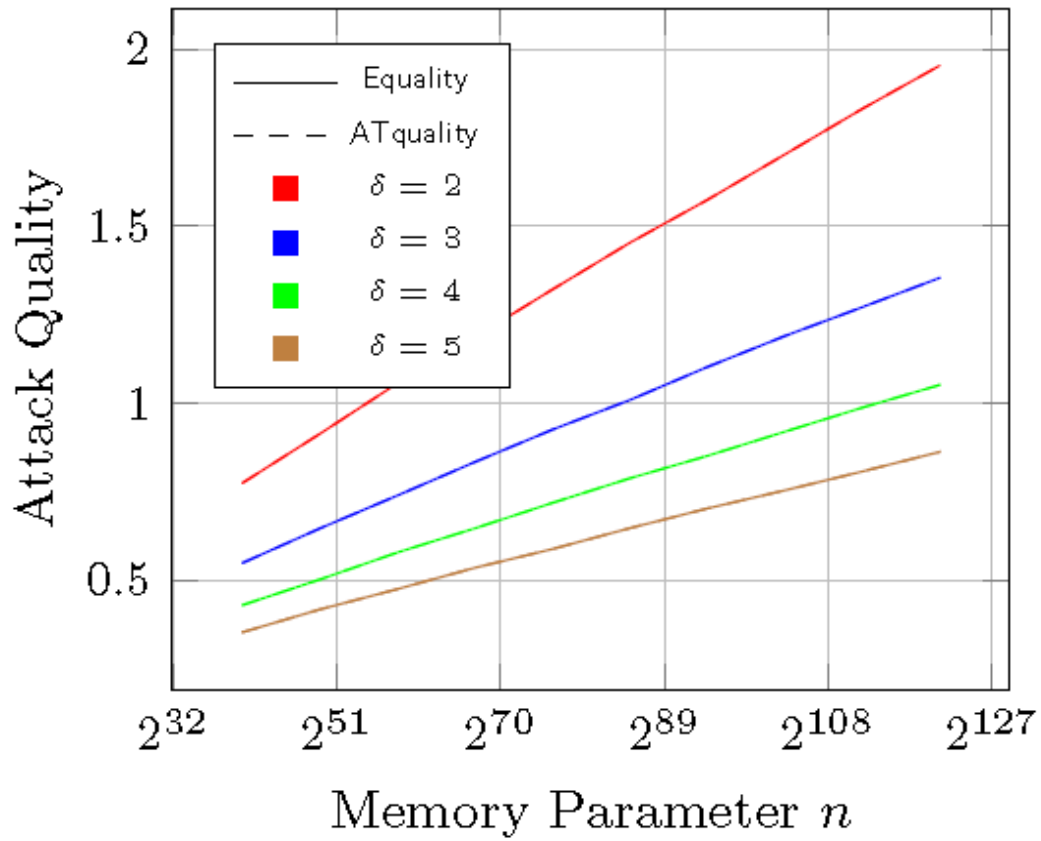
- **Thm:** If G has n nodes and constant in-degree then:

$$\forall \varepsilon > 0 \quad \mathrm{E_R}(G) = o\left(\frac{n^2}{\log(n)^{1-\varepsilon}} + nR\right)$$

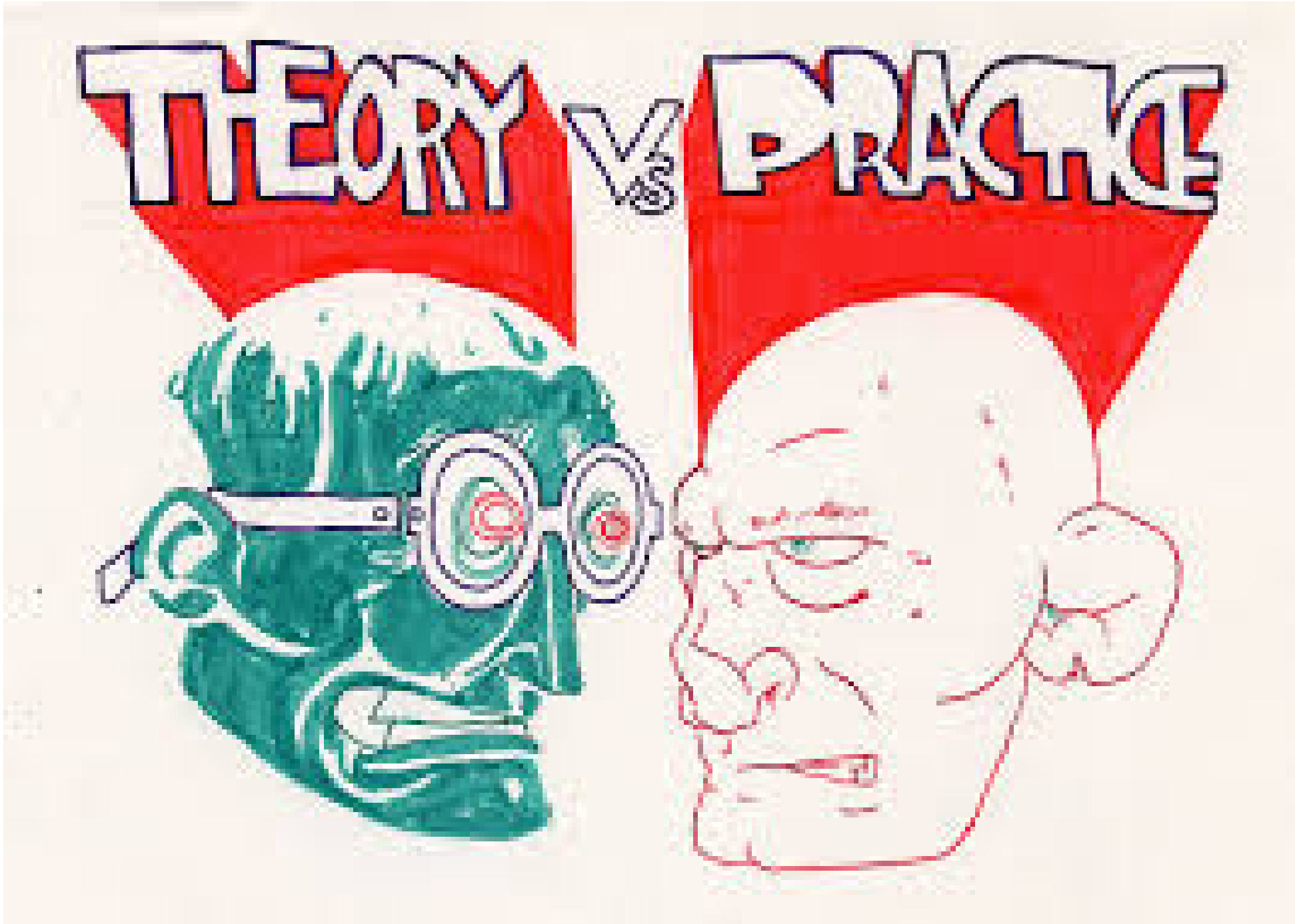
Practical Consequences (R = 3,000)



(a) Argon2i and SB



(b) Ideal iMHF


THEORY
VS
PRACTICE

Drama: Are the attacks `Practical'

- Argon2i team: No, at least for reali
- Recent: Argon2i-B submitted to IRT Task Force) for standardization.
- New Result [AB16b]:
  - New heuristics to reduce overhead by constant factor
  - Simulate the attack on real instances

New Simulation Results : [AB16b]

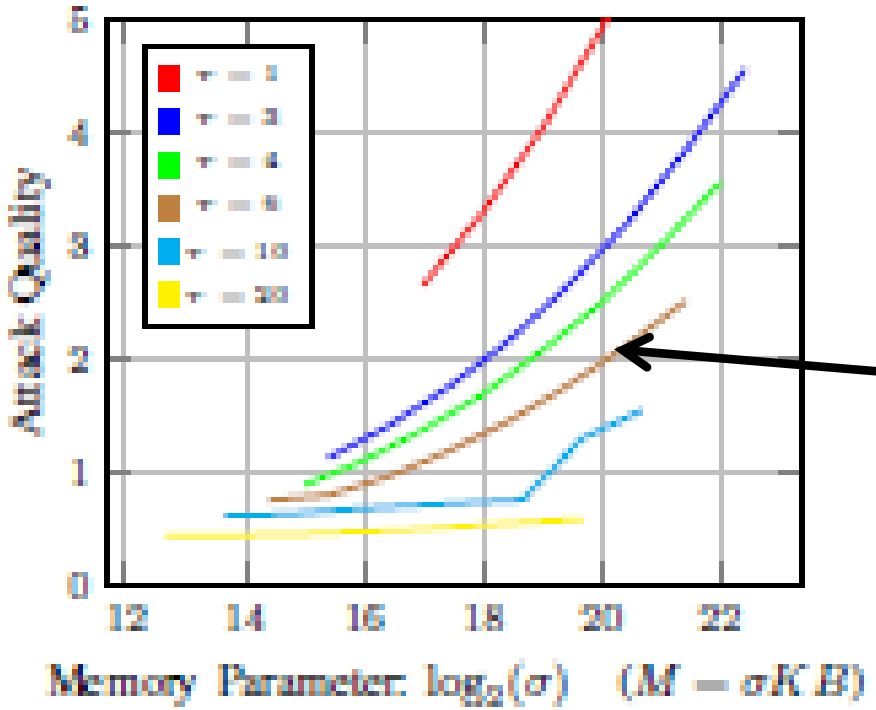


Figure 1: Argon2i-B Attack Quality

Parameter setting could easily be chosen when following Argon2i-B guidelines

Pessimistic Argon 2i-B parameter

Attack on Argon 2i-B is practical even for pessimistic parameter ranges (brown line).

Outline

- Motivation
- Data Independent Memory Hard Functions (iMHFs)
- Attacks
- **Constructing iMHFs (New!)**
  - **Depth-Robustness is *sufficient***
- Conclusions and Open Questions

Depth-Robustness is Sufficient! [ABP16]

Key Theorem: Let G=(V,E) be (e,d)-depth robust then CC(G)$\geq ed$.

Implications: There exists a constant indegree graph G with

$$\mathrm{CC(G)} \geq \Omega\left(\frac{n^2}{\log n}\right).$$

Previous Best [AS15]: $\Omega\left(\frac{n^2}{\log^{10} n}\right)$

[AB16]: For all constant indegree graphs $\mathrm{CC(G)} = O\left(\frac{n^2 \log\log n}{\log n}\right)$.

Depth-Robustness is Sufficient! [ABP16]

Proof: Let $P_1,\ldots P_t$ denote an (optimal) pebbling of G. For 0< i < d define

$$S_i = P_i \cup P_{d+i} \cup P_{2d+i} \cup \cdots$$

one of the sets S_i has size at most CC(G)/d. Now we claim that

$$d \geq \text{depth}(G\text{-}S_i)$$

because any path in G-S_i must have been completely pebbled at some point. Thus, it must have been pebbled entirely during some interval of length d. Thus, G (CC(G)/d,d)-reducible. It follows that CC(G)$\geq ed$.

Proof by Picture

$$S_i = P_i \cup P_{d+i} \cup P_{2d+i} \cup \cdots$$

$P_1, P_2, \ldots, P_{i-1}, P_i, P_{i+1}, \ldots, P_{i+d-1}, P_{i+d}, P_{i+d+1}, \ldots, P_{i+2d-1}, P_{i+2d}, \ldots$

d rounds d rounds

Claim: $|S_i| \geq e$

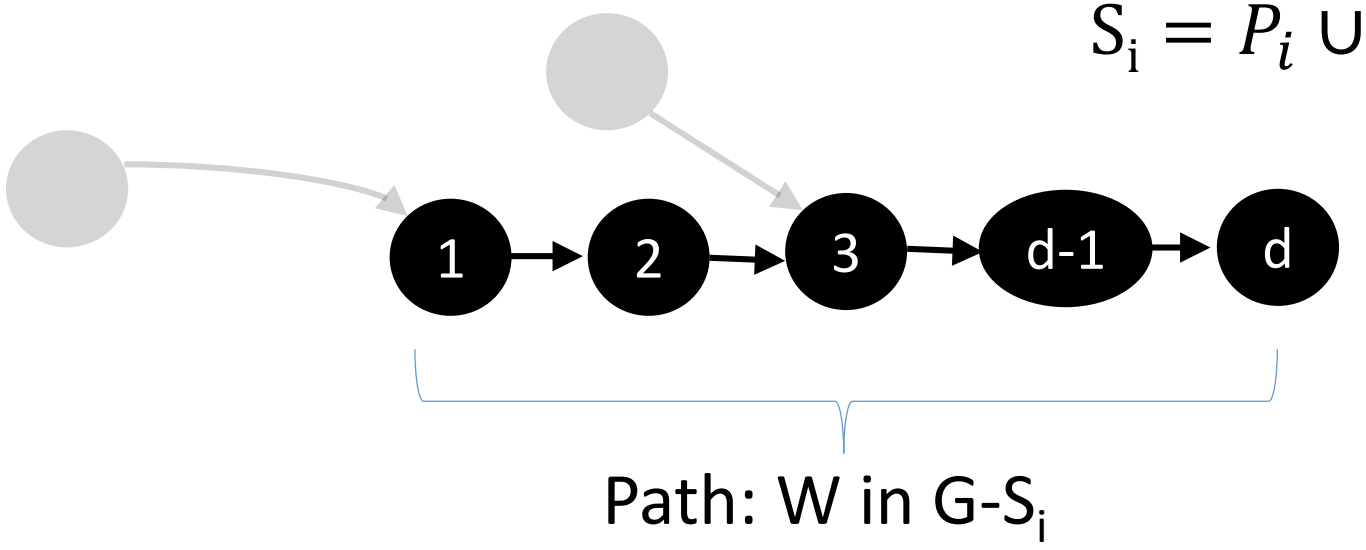
Implication

Claim: $|S_i| \geq e$

$$CC(G) \geq \sum_{t} |P_t| \geq \sum_{i=1}^{d} |S_i| \geq \sum_{i=1}^{d} e \geq ed$$

Contradiction by Picture

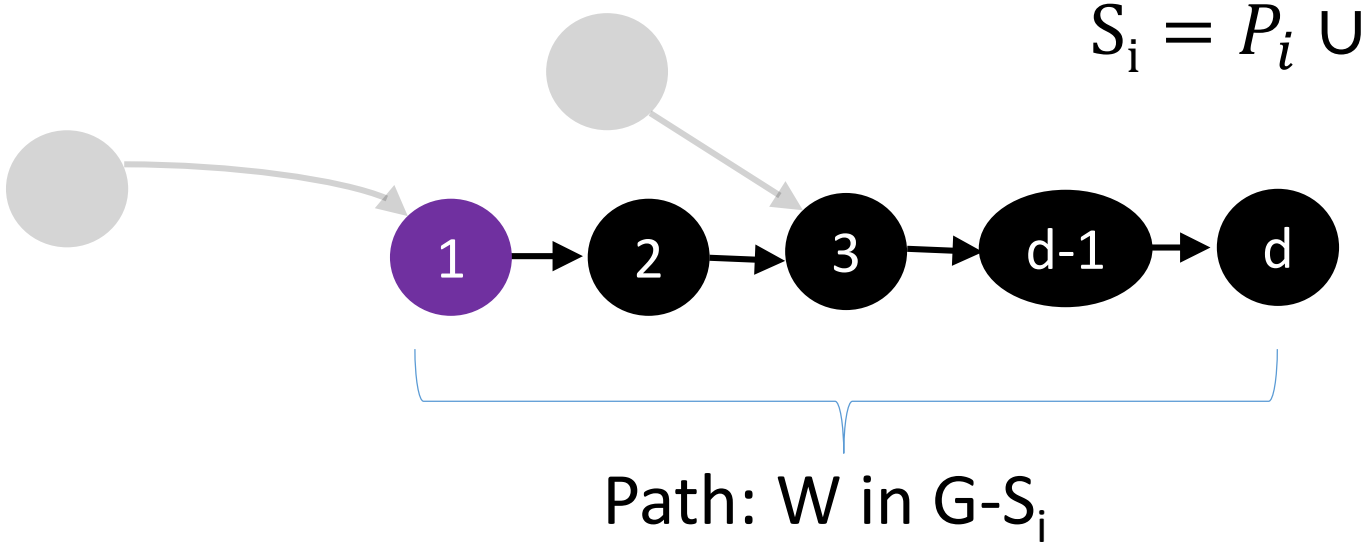
$$S_i = P_i \cup P_{d+i} \cup P_{2d+i} \cup \cdots$$



Step i: W contains no pebbles since $P_i \subset S_i$

Contradiction by Picture

$$S_i = P_i \cup P_{d+i} \cup P_{2d+i} \cup \cdots$$

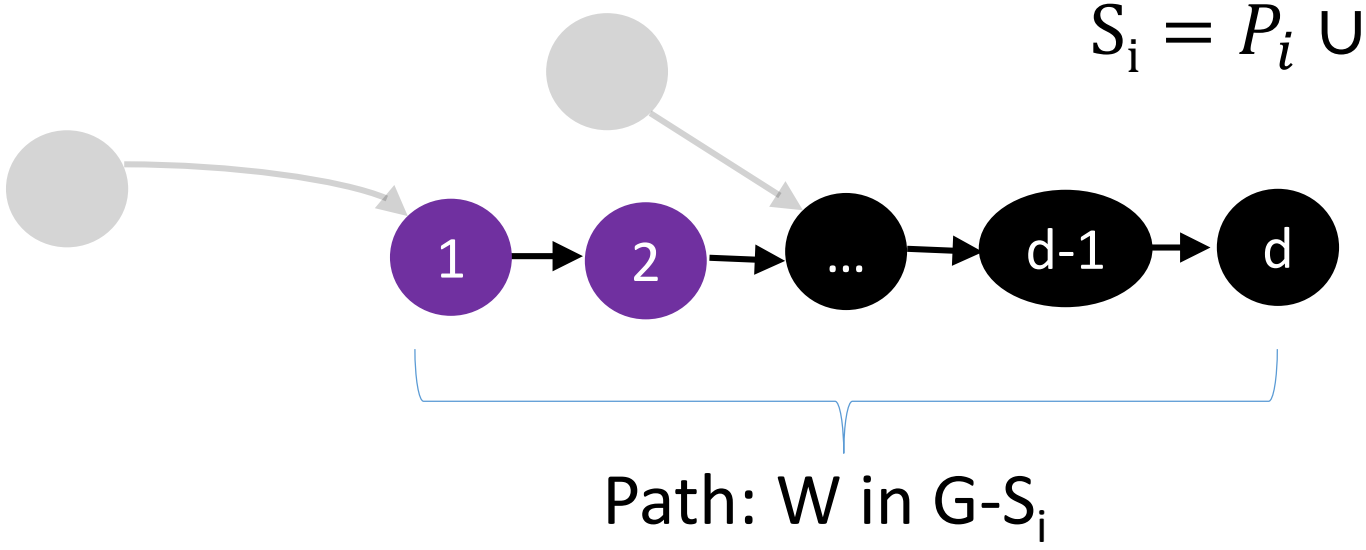


Step i: W contains no pebbles since $P_i \subset S_i$

Step i+1: W-{1} contains no pebbles

Contradiction by Picture

$$S_i = P_i \cup P_{d+i} \cup P_{2d+i} \cup \cdots$$



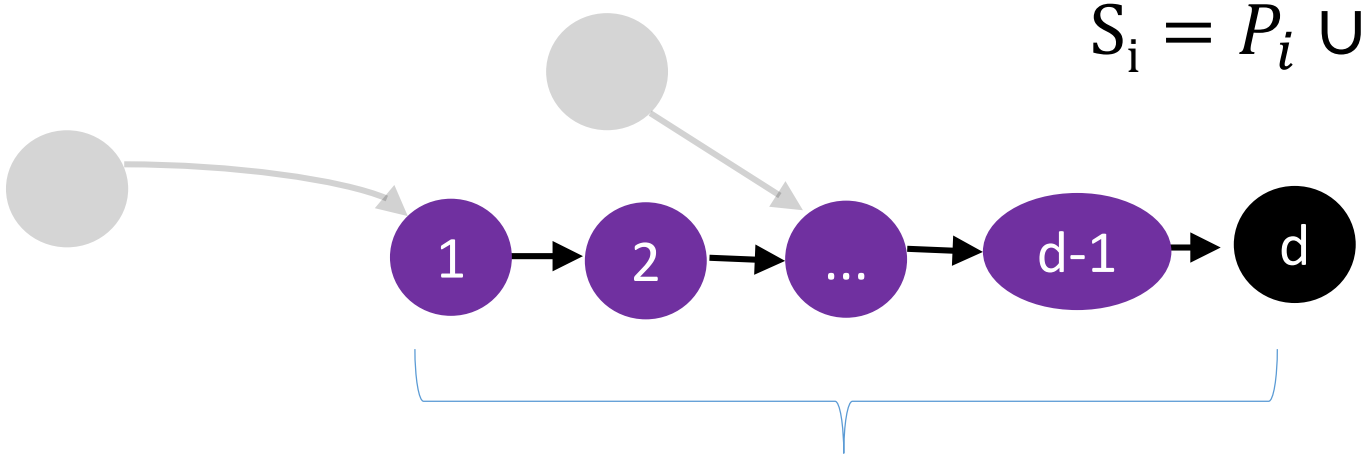
Step i: W contains no pebbles since $P_i \subset S_i$

Step i+1: W-{1} contains no pebbles

Step i+2: W-{1,2} contains no pebbles

Contradiction by Picture

$$S_i = P_i \cup P_{d+i} \cup P_{2d+i} \cup \cdots$$



Path: W in G-S_i

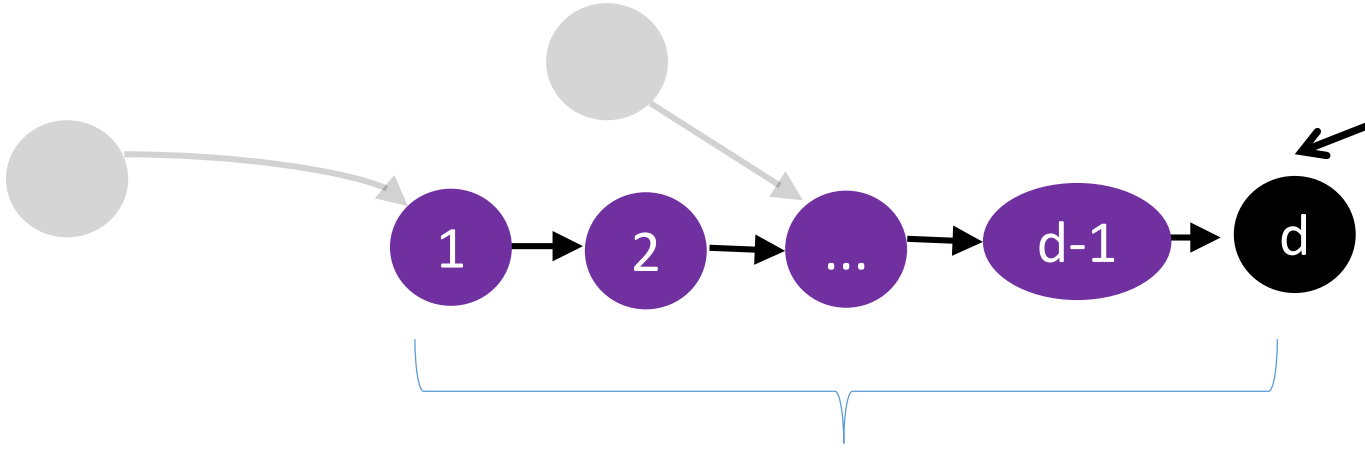
Step i: W contains no pebbles since $P_i \subset S_i$

Step i+1: W-{1} contains no pebbles

Step i+2: W-{1,2} contains no pebbles

Step i+d-1: W-{1,…,d-1} contains no pebbles

Contradiction by Picture



Contradiction!
d was never pebbled.

Path: W in G-S_i

$\therefore |S_i| \geq e$

Step i+d: W contains no pebbles since $P_{i+d} \subset S_i$

Step i+d-1: W-{1,…,d-1} contains no pebbles

Positive Result: Consequences

Theorem [ABP16]: Let G=(V,E) be (e,d)-depth robust then $\mathrm{E}_{\mathrm{R}}(G) \geq ed$.

Theorem[EGS75]: There is an $\left(\Omega(n), \Omega(n)\right)$-depth robust DAG G with indegree $\delta = O(\log n)$.

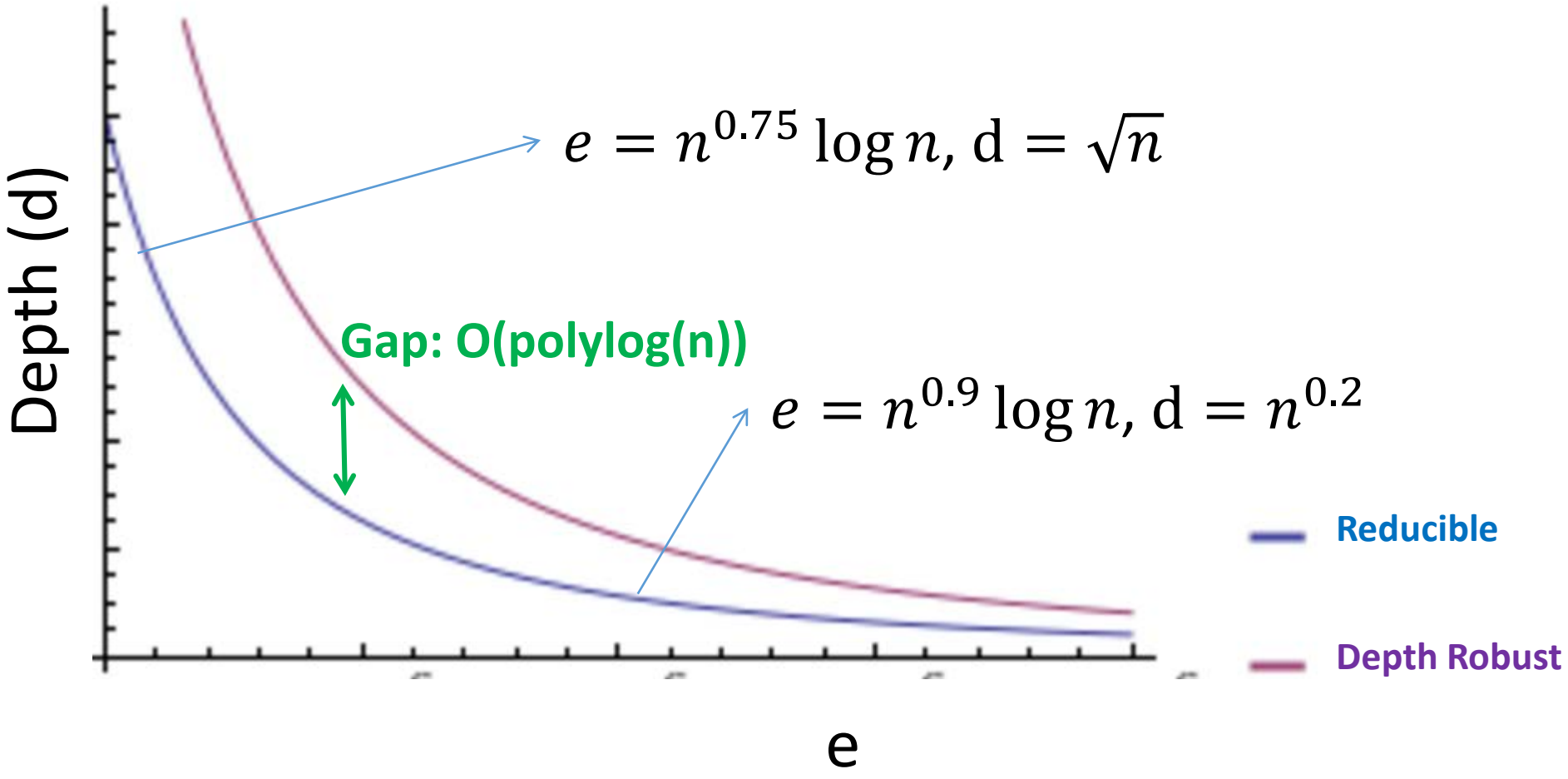
Theorem [ABP16] There is a DAG G with maximum indegree $\delta = 2$ and $\mathrm{E}_{\mathrm{R}}(G) = \Omega\left(\frac{n^2}{\log n}\right)$. Furthermore, there is a sequential pebbling algorithm N with cost $\mathrm{E}_{\mathrm{R}}(N) = O\left(\frac{n^2}{\log n}\right)$.

More New Results

MHF	Upper Bound	Lower Bound
Argon2i-A	$\tilde{O}(n^{1.71})$ [ABP16] ~~$\tilde{O}(n^{1.75})$~~ [This work]	$\tilde{\Omega}(n^{1.66})$ [ABP16]
Catena	$\tilde{O}(n^{1.618})$ [ABP16] ~~$O(n^{1.67})$~~ [This work]	$\tilde{\Omega}(n^{1.5})$ [ABP16]
SCRYPT (data dependent)	$O(n^2)$ [Naïve, P12]	$\Omega(n^2)$ [ACPRT16]

Idea: Apply our attack recursively during balloon phases

(e,d)-reducible curve for Argon2i-A



Recursive Attack

$$CC(G) \leq en + \frac{n}{e} nd$$

$$CC(G) \leq en + \frac{n}{e} CC(G')$$

$$CC(G) \leq e_1 n + \frac{n}{e_1}\left(e_2 d_1 + \frac{n}{e_2} d_2 n\right)$$

....

Conclusions

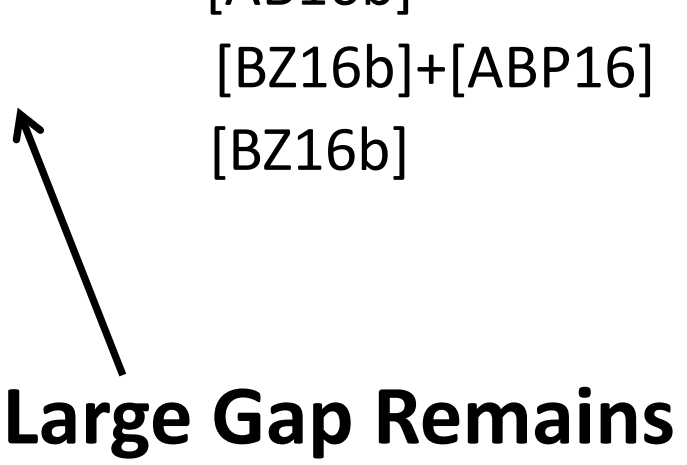
- Depth-robustness is a necessary and sufficient for secure iMHFs
  - [AB16] [ABP16]

- Big Challenge: Improved Constructions of Depth-Robust Graphs
  - We already have constructions in theory [EGS77, PR80, …]
  - But constants matter!

More Open Questions

- Computational Complexity of Pebbling
  - NP-Hard to determine CC(G) [BZ16]
  - Hardness of Approximation?

- What is CC(Argon2i-B)?
  - Upper Bound: $O(n^{1.8})$ [AB16b]
  - Recursive attack: $O(n^{1.77})$ [BZ16b]+[ABP16]
  - Lower Bound: $\Omega(n^{1.66})$ [BZ16b]



Large Gap Remains

Thanks for Listening