Detecting Feedback Vertex Sets of Size k in O*(2.7^k) Time

Jason Li

With Jesper Nederlof (TU Eindhoven)

SODA 2020

January 6, 2020

Introduction

Def: given a graph G, a feedback vertex set (FVS) is a set F of vertices s.t. G-F is a forest

Equivalently, F hits all cycles of G

Introduction

Def: given a graph G, a feedback vertex set (FVS) is a set F of vertices s.t. G-F is a forest

Equivalently, F hits all cycles of G

Parameterized by k: given a graph G, find a FVS of size k in G or determine that none exist

Introduction

Def: given a graph G, a feedback vertex set (FVS) is a set F of vertices s.t. G-F is a forest

Equivalently, F hits all cycles of G

Parameterized by k: given a graph G, find a FVS of size k in G or determine that none exist

Want time FPT in k: f(k)*poly(n)

Goal in FPT setting: minimize function f(k).

poly(n) factor does not matter

Prior Work

Downey and Fellows '92: $f(k) = k^{O(k)}$

Becker et al. [BBG'00]: f(k)=4^k, randomized

Cygan et al. [CNP+'11]: f(k)=3^k, randomized

- actually runs in 3^tw time, given a tree decomposition of width tw

Kociumaka and Pilipczuk '14: f(k)=3.62^k deterministic

Prior Work

Downey and Fellows ‘92: $f(k) = k^{O(k)}$

Becker et al. [BBG’00]: f(k)=4^k, randomized

Cygan et al. [CNP+’11]: f(k)=3^k, randomized

- actually runs in 3^tw time, given a tree decomposition of width tw

Kociumaka and Pilipczuk ‘14: f(k)=3.62^k deterministic

Our result: f(k)=2.7^k randomized.

Conceptual message: 3^k barrier can be broken

Combines techniques from [BBG’00] and [CNP+11].

Prior Work

Downey and Fellows ‘92: $f(k) = k^{O(k)}$

Becker et al. [BBG’00]: f(k)=4^k, randomized

Cygan et al. [CNP+’11]: f(k)=3^k, randomized

- actually runs in 3^tw time, given a tree decomposition of width tw

Kociumaka and Pilipczuk ‘14: f(k)=3.62^k deterministic

Our result: f(k)=2.7^k randomized.

Conceptual message: 3^k barrier can be broken

Combines techniques from [BBG’00] and [CNP+11].

This talk: (3-ε)^k, or how to break 3^k.

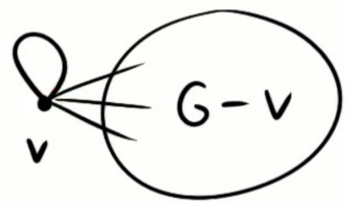
Common Reductions

- Reduction Rule: (G, k) —> (G′, k′) where G has size-k FVS iff G′ has size-k′ FVS

Common Reductions

- Reduction Rule: (G, k) —> (G′, k′) where G has size-k FVS iff G′ has size-k′ FVS

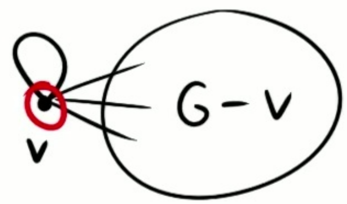
(1) Self-loop at v:



Common Reductions

- Reduction Rule: (G, k) —> (G', k') where G has size-k FVS iff G' has size-k' FVS

(1) Self-loop at v:



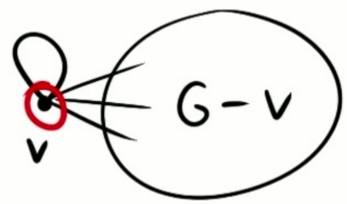
Must select v in FVS

(G, k) —> (G-v, k-1)

Common Reductions

- Reduction Rule: (G, k) —> (G', k') where G has size-k FVS iff G' has size-k' FVS

(1) Self-loop at v:



Must select v in FVS

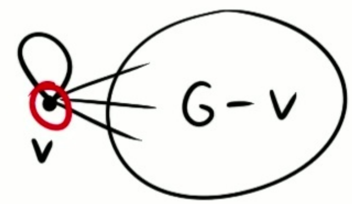
(G, k) —> (G-v, k-1)

Remove v and decrease k by 1

Common Reductions

- Reduction Rule: (G, k) —> (G′, k′) where G has size-k FVS iff G′ has size-k′ FVS

(1) Self-loop at v:

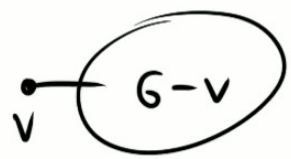


Must select v in FVS

(G, k) —> (G-v, k-1)

Remove v and decrease k by 1

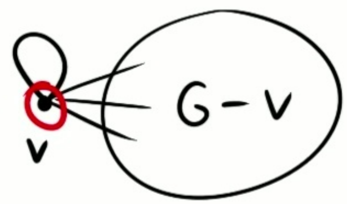
(2) Leaf vertex v (degree 1):



Common Reductions

- Reduction Rule: (G, k) —> (G', k') where G has size-k FVS iff G' has size-k' FVS

(1) Self-loop at v:

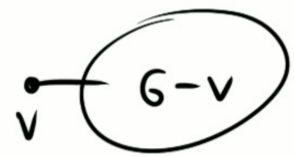


Must select v in FVS

(G, k) —> (G-v, k-1)

Remove v and decrease k by 1

(2) Leaf vertex v (degree 1):



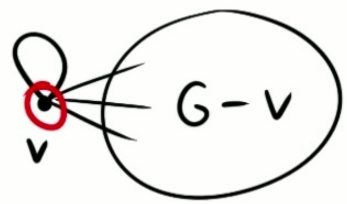
v doesn't belong in any cycle

Remove v: (G, k) —> (G-v, k)

Common Reductions

- Reduction Rule: (G, k) —> (G′, k′) where G has size-k FVS iff G′ has size-k′ FVS

(1) Self-loop at v:

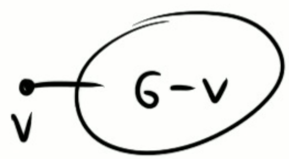


Must select v in FVS

(G, k) —> (G-v, k-1)

Remove v and decrease k by 1

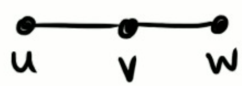
(2) Leaf vertex v (degree 1):



v doesn’t belong in any cycle

Remove v: (G, k) —> (G-v, k)

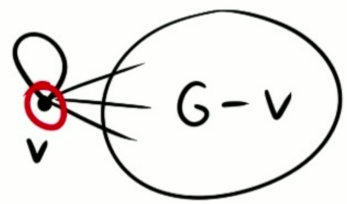
(3) Degree-2 vertex v:



Common Reductions

- Reduction Rule: (G, k) —> (G', k') where G has size-k FVS iff G' has size-k' FVS

(1) Self-loop at v:

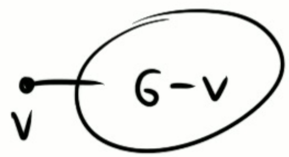


Must select v in FVS

(G, k) —> (G-v, k-1)

Remove v and decrease k by 1

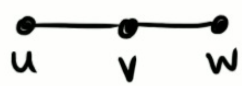
(2) Leaf vertex v (degree 1):



v doesn't belong in any cycle

Remove v: (G, k) —> (G-v, k)

(3) Degree-2 vertex v:



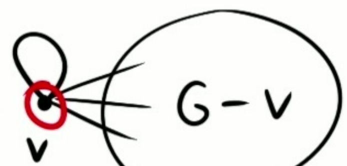
Any cycle containing v also contains u

If select v in FVS, then might as well select u instead

Common Reductions

- Reduction Rule: (G, k) —> (G′, k′) where G has size-k FVS iff G′ has size-k′ FVS

(1) Self-loop at v:

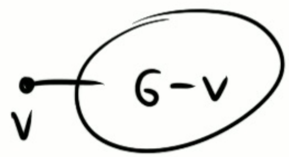


Must select v in FVS

(G, k) —> (G-v, k-1)

Remove v and decrease k by 1

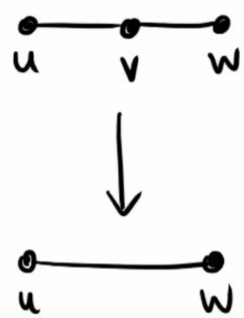
(2) Leaf vertex v (degree 1):



v doesn't belong in any cycle

Remove v: (G, k) —> (G-v, k)

(3) Degree-2 vertex v:



Any cycle containing v also contains u

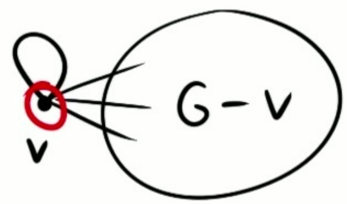
If select v in FVS, then might as well select u instead

Delete v and add edge (u,w)

Common Reductions

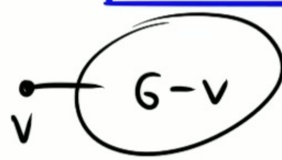
- Reduction Rule: (G, k) —> (G', k') where G has size-k FVS iff G' has size-k' FVS

(1) Self-loop at v:



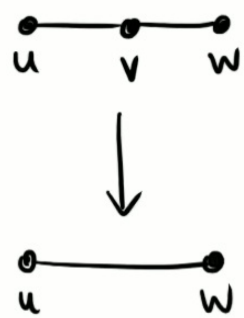
When (1),(2),(3) no longer apply:
- no self-loops
- minimum degree 3

(2) Leaf vertex v (degree 1):



v doesn't belong in any cycle
Remove v: (G, k) —> (G-v, k)

(3) Degree-2 vertex v:



Any cycle containing v also contains u
If select v in FVS, then might as well select u instead

Delete v and add edge (u,w)

Becker et al. Reduction

- Reduction Rule: (G, k) —> (G′, k′) where G has size-k FVS iff G′ has size-k′ FVS

Becker et al. Reduction

- Reduction Rule: (G, k) —> (G', k') where G has size-k FVS iff G' has size-k' FVS
- Probabilistic Reduction: (G, k) —> (G', k') where with probability p, G has size-k FVS iff G' has size-k' FVS

Becker et al. Reduction

- Reduction Rule: (G, k) —> (G', k') where G has size-k FVS iff G' has size-k' FVS
- Probabilistic Reduction: (G, k) —> (G', k') where with probability p, G has size-k FVS iff G' has size-k' FVS

Becker et al. [BBG'00]: sample v proportional to deg(v)

Becker et al. Reduction

- Reduction Rule: (G, k) —> (G', k') where G has size-k FVS iff G' has size-k' FVS
- **Probabilistic** Reduction: (G, k) —> (G', k') where **with probability p**, G has size-k FVS iff G' has size-k' FVS

Becker et al. [BBG'00]: sample v proportional to deg(v)

Add v to FVS: (G, k) -> (G-v, k-1)

Becker et al. Reduction

- Reduction Rule: (G, k) —> (G', k') where G has size-k FVS iff G' has size-k' FVS
- Probabilistic Reduction: (G, k) —> (G', k') where with probability p, G has size-k FVS iff G' has size-k' FVS

Becker et al. [BBG'00]: sample v proportional to deg(v)

Add v to FVS: (G, k) -> (G-v, k-1)

Theorem: if G has minimum degree ≥3 and a FVS of size k, then with probability ≥1/4, v is in the FVS

Becker et al. Reduction

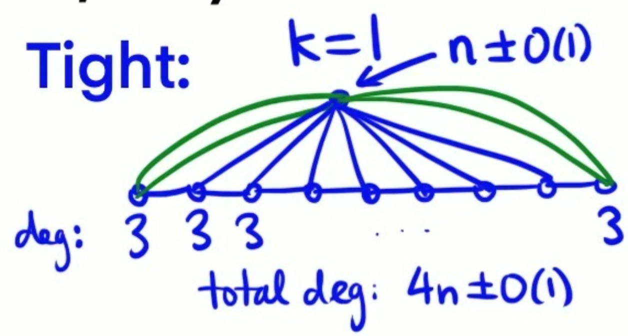
- Reduction Rule: (G, k) —> (G', k') where G has size-k FVS iff G' has size-k' FVS
- Probabilistic Reduction: (G, k) —> (G', k') where with probability p, G has size-k FVS iff G' has size-k' FVS

Becker et al. [BBG'00]: sample v proportional to deg(v)

Add v to FVS: (G, k) -> (G-v, k-1)

Theorem: if G has minimum degree ≥3 and a FVS of size k, then with probability ≥1/4, v is in the FVS

Tight:



Becker et al. Reduction

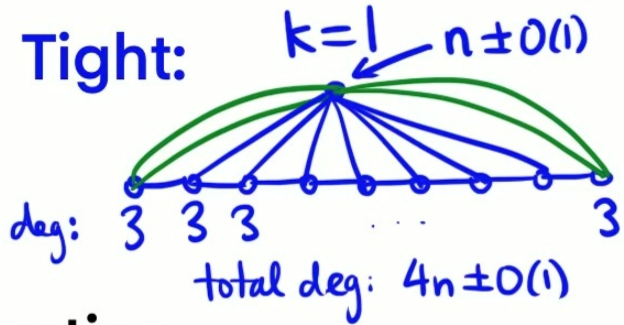
- Reduction Rule: (G, k) —> (G', k') where G has size-k FVS iff G' has size-k' FVS
- Probabilistic Reduction: (G, k) —> (G', k') where with probability p, G has size-k FVS iff G' has size-k' FVS

Becker et al. [BBG'00]: sample v proportional to deg(v)

Add v to FVS: (G, k) -> (G-v, k-1)

Theorem: if G has minimum degree ≥3 and a FVS of size k, then with probability ≥1/4, v is in the FVS

Tight:



Prob. 1/4 to decrease k by 1 and preserve reduction

=> prob. 1/4^k to go all the way. Repeat 4^k times: O*(4^k) algo.

Our Approach

Dense case: $m >> O(k)$:

Sparse case: $m \leq O(k)$:

Our Approach

Dense case: $m >> O(k)$:

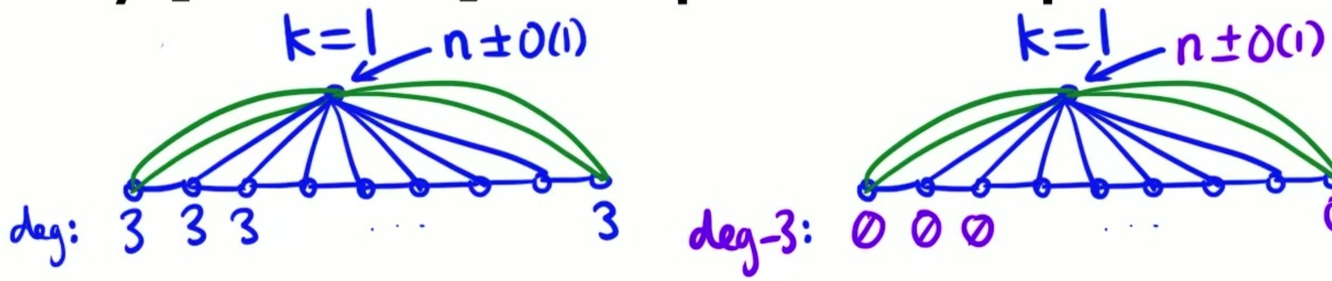
- Modify [BBG'00]: sample v with prob. $\sim deg(v)-3$

Sparse case: $m \leq O(k)$:

Our Approach

Dense case: m >> O(k):

- Modify [BBG'00]: sample v with prob. ~ deg(v)-3



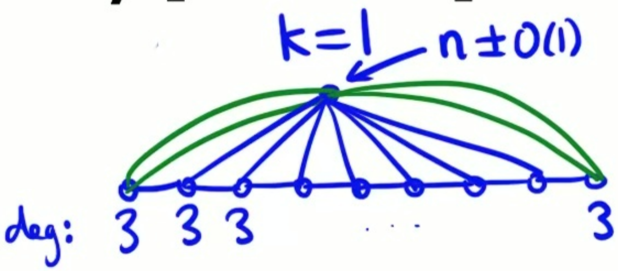
Success w.p. 1

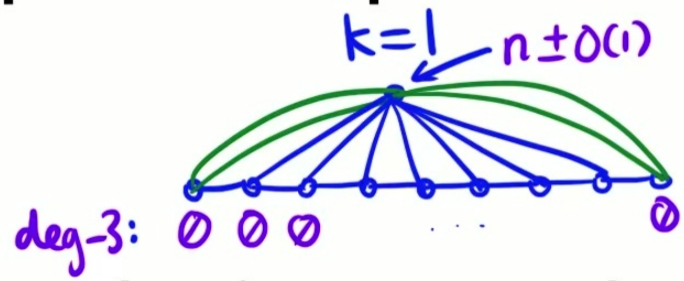
Sparse case: m ≤ O(k):

Our Approach

Dense case: $m >> O(k)$:

- Modify [BBG'00]: sample v with prob. ~ deg(v)-3





Success w.p. 1

Lemma: if $m \geq 100k$, then reduction succeeds w.p. 1/2.99

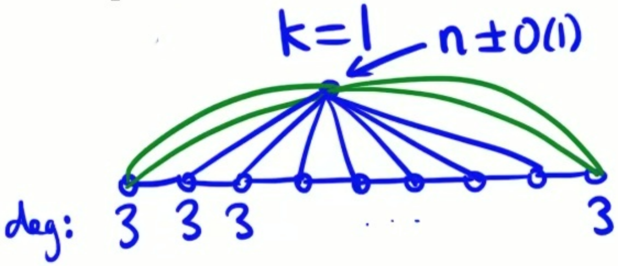
Repeat k times => 2.99^k algo

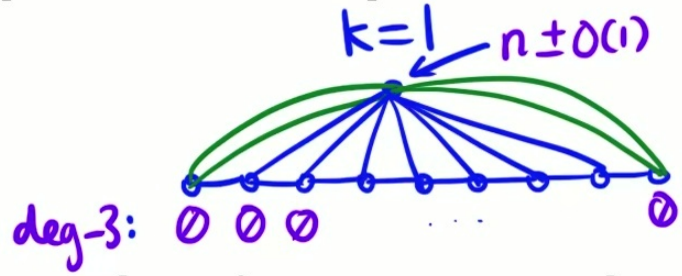
Sparse case: $m \leq O(k)$:

Our Approach

Dense case: m >> O(k):

- Modify [BBG'00]: sample v with prob. ~ deg(v)-3





Success w.p. 1

Lemma: if m ≥ 100k, then reduction succeeds w.p. 1/2.99

Repeat k times => 2.99^k algo

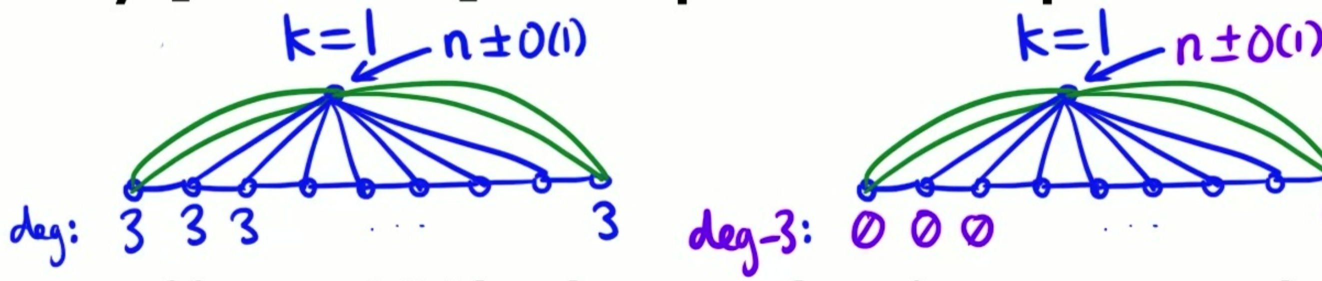
Sparse case: m ≤ O(k):

Lemma: if m ≤ 100k and exists FVS size k, then G has treewidth (1-Ω(1))k

Our Approach

Dense case: m >> O(k):

- Modify [BBG'00]: sample v with prob. ~ deg(v)-3



Success w.p. 1

Lemma: if m ≥ 100k, then reduction succeeds w.p. 1/2.99

Repeat k times => 2.99^k algo

Sparse case: m ≤ O(k):

Lemma: if m ≤ 100k and exists FVS size k, then

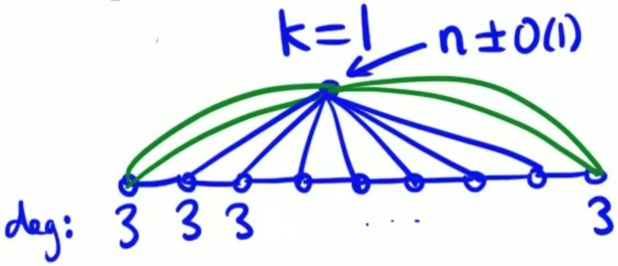
G has treewidth (1-Ω(1))k

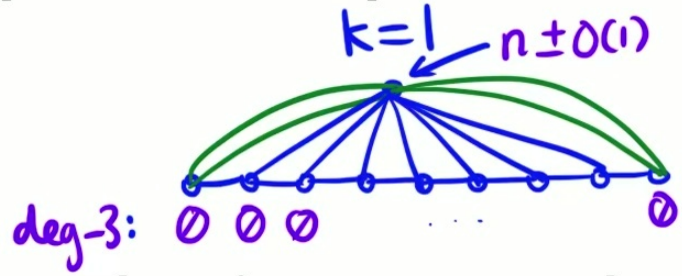
Can find such a tree decomposition. Run 3^tw algo [CNP+11]

Our Approach

Dense case: m >> O(k):

- Modify [BBG'00]: sample v with prob. ~ deg(v)-3





Success w.p. 1

Lemma: if m ≥ 100k, then reduction succeeds w.p. 1/2.99

Repeat k times => 2.99^k algo

Sparse case: m ≤ O(k):

Lemma: if m ≤ 100k and exists FVS size k, then

G has treewidth **(1-Ω(1))k** ⇒ time $3^{(1-\Omega(1))k} = (3-\varepsilon)^k$.

Can find such a tree decomposition. Run 3^tw algo [CNP+11]

Dense case: $m >> O(k)$

Lemma: Let F be a FVS of graph G. Then,

$$\deg(V\setminus F) \leq \deg(F) + 2(|V\setminus F|-1), \quad \text{where } \deg(S) := \sum_{v\in S}\deg(v)$$

Dense case: $m >> O(k)$

Lemma: Let F be a FVS of graph G. Then,

$$\deg(V\setminus F) \leq \deg(F) + 2(|V\setminus F|-1), \quad \text{where } \deg(S) := \sum_{v\in S}\deg(v)$$

Proof:

- $G-F$ is forest => at most $|V\setminus F|-1$ edges with both endpoints in $G-F$.
  Each contributes +2 to $\deg(V\setminus F)$, for a total of $\leq 2(|V\setminus F|-1)$
- Every edge between F and $V\setminus F$ contributes +1 to both $\deg(F)$ and $\deg(V\setminus F)$

Dense case: $m >> O(k)$

Lemma: Let F be a FVS of graph G. Then,

$$\deg(V\setminus F) \leq \deg(F) + 2(|V\setminus F|-1), \quad \text{where } \deg(S) := \sum_{v\in S} \deg(v)$$

Proof:

- $G-F$ is forest => at most $|V\setminus F|-1$ edges with both endpoints in $G-F$. Each contributes +2 to $\deg(V\setminus F)$, for a total of $\leq 2(|V\setminus F|-1)$
- Every edge between F and $V\setminus F$ contributes +1 to both $\deg(F)$ and $\deg(V\setminus F)$

If $n \geq 4k$ and F has size k:

Dense case: m >> O(k)

Lemma: Let F be a FVS of graph G. Then,

$$\deg(V\setminus F) \leq \deg(F) + 2(|V\setminus F|-1), \quad \text{where } \deg(S) := \sum_{v\in S}\deg(v)$$

Proof:

- G-F is forest => at most $|V\setminus F|-1$ edges with both endpoints in G-F.
  Each contributes +2 to $\deg(V\setminus F)$, for a total of $\leq 2(|V\setminus F|-1)$
- Every edge between F and V\F contributes +1 to both deg(F) and deg(V\F)

If $n \geq 4k$ and F has size k:

$$\begin{aligned}\sum_{v\notin F}(\deg(v)-3) &= \deg(V\setminus F)-3|V\setminus F| \\ &\leq \deg(F)+2(|V\setminus F|-1)-3|V\setminus F| \\ &\leq \deg(F)-3|V\setminus F|\end{aligned}$$

$$n\geq 4k: \quad |V\setminus F| \geq 3k = 3|F|$$

$$\leq \deg(F) - 3|F| = \sum_{v\in F}(\deg(v)-3)$$

Dense case: m >> O(k)

Lemma: Let F be a FVS of graph G. Then,

$$\deg(V\setminus F) \le \deg(F) + 2(|V\setminus F|-1), \quad \text{where } \deg(S) := \sum_{v\in S}\deg(v)$$

Proof:

- G-F is forest => at most $|V\setminus F|-1$ edges with both endpoints in G-F.
  Each contributes +2 to $\deg(V\setminus F)$, for a total of $\le 2(|V\setminus F|-1)$
- Every edge between F and $V\setminus F$ contributes +1 to both $\deg(F)$ and $\deg(V\setminus F)$

If $n \ge 4k$ and F has size k:

$$\begin{aligned}
\sum_{v\notin F}(\deg(v)-3) &= \deg(V\setminus F) - 3|V\setminus F| \\
&\le \deg(F) + 2(|V\setminus F|-1) - 3|V\setminus F| \\
&\le \deg(F) - 3|V\setminus F|
\end{aligned}$$

$n \ge 4k$: $|V\setminus F| \ge 3k = 3|F|$

$$\le \deg(F) - 3|F| = \sum_{v\in F}(\deg(v)-3)$$

choose $v\in F$ w.p. $\ge 1/2$

Dense case: m >> O(k)

Lemma: Let F be a FVS of graph G. Then,

$$\deg(V\setminus F) \le \deg(F) + 2(|V\setminus F|-1), \quad \text{where } \deg(S) := \sum_{v\in S}\deg(v)$$

Proof:

- G-F is forest => at most $|V\setminus F|-1$ edges with both endpoints in G-F. Each contributes +2 to $\deg(V\setminus F)$, for a total of $\le 2(|V\setminus F|-1)$
- Every edge between F and V\F contributes +1 to both deg(F) and deg(V\F)

If $n \ge 4k$ and F has size k:

$$\sum_{v\notin F}(\deg(v)-3) = \deg(V\setminus F) - 3|V\setminus F|$$

$$\le \deg(F) + 2(|V\setminus F|-1) - 3|V\setminus F|$$

$$\le \deg(F) - 3|V\setminus F|$$

$n \ge 4k$: $|V\setminus F| \ge 3k = 3|F|$

$$\le \deg(F) - 3|F| = \sum_{v\in F}(\deg(v)-3)$$

$\le$

choose $v\in F$ w.p. $\ge 1/2$

If $m \ge 20n$ and F has size k:

Dense case: m >> O(k)

Lemma: Let F be a FVS of graph G. Then,

$$\deg(V\setminus F) \le \deg(F) + 2(|V\setminus F|-1), \quad \text{where } \deg(S) := \sum_{v\in S}\deg(v)$$

Proof:

- G-F is forest => at most $|V\setminus F|-1$ edges with both endpoints in G-F. Each contributes +2 to $\deg(V\setminus F)$, for a total of $\le 2(|V\setminus F|-1)$
- Every edge between F and $V\setminus F$ contributes +1 to both $\deg(F)$ and $\deg(V\setminus F)$

If $n \ge 4k$ and F has size k:

$$\underline{\sum_{v\notin F}(\deg(v)-3)} = \deg(V\setminus F) - 3|V\setminus F|$$
$$\le \deg(F) + 2(|V\setminus F|-1) - 3|V\setminus F|$$
$$\le \deg(F) - 3|V\setminus F|$$

$n \ge 4k$: $|V\setminus F| \ge 3k = 3|F|$

$$\le \deg(F) - 3|F| = \underline{\sum_{v\in F}(\deg(v)-3)}$$

choose $v \in F$ w.p. $\ge 1/2$

If $m \ge 20n$ and F has size k:

$$\underbrace{\deg(V)}_{2m} - \deg(F) \le \deg(F) + 2(\underbrace{|V\setminus F|}_{n-k} - 1)$$

$$\deg(F) \ge \frac{1}{2}(2m - 2(n-k-1)) \ge m - n + k$$

$$\sum_{v\in F}(\deg(v)-3) = \deg(v) - 3|F| \ge m - 4n \ge 0.8m$$

$$\sum_{v\in V}(\deg(v)-3) \le \deg(V) = 2m$$

Dense case: m >> O(k)

Lemma: Let F be a FVS of graph G. Then,

$$\deg(V\setminus F) \leq \deg(F) + 2(|V\setminus F|-1), \quad \text{where } \deg(S) := \sum_{v\in S}\deg(v)$$

Proof:

- G-F is forest => at most $|V\setminus F|-1$ edges with both endpoints in G-F. Each contributes +2 to $\deg(V\setminus F)$, for a total of $\leq 2(|V\setminus F|-1)$
- Every edge between F and $V\setminus F$ contributes +1 to both $\deg(F)$ and $\deg(V\setminus F)$

If $n \geq 4k$ and F has size k:

$$\underline{\sum_{v\notin F}(\deg(v)-3)} = \deg(V\setminus F) - 3|V\setminus F|$$
$$\leq \deg(F) + 2(|V\setminus F|-1) - 3|V\setminus F|$$
$$\leq \deg(F) - 3|V\setminus F|$$

$n \geq 4k$: $|V\setminus F| \geq 3k = 3|F|$

$$\leq \deg(F) - 3|F| = \underline{\sum_{v\in F}(\deg(v)-3)}$$

choose $v\in F$ w.p. $\geq 1/2$

If $m \geq 20n$ and F has size k:

$$\underbrace{\deg(V)}_{2m} - \deg(F) \leq \deg(F) + 2(\underbrace{|V\setminus F|}_{n-k}-1)$$

$$\deg(F) \geq \frac{1}{2}(2m - 2(n-k-1)) \geq m-n+k$$

$$\sum_{v\in F}(\deg(v)-3) = \deg(v) - 3|F| \geq m - 4n \geq \underline{0.8m}$$

$$\sum_{v\in V}(\deg(v)-3) \leq \deg(V) = \underline{2m} \qquad \frac{0.8m}{2m} \Rightarrow \text{success w.p. } \geq 0.4$$

Dense case: m >> O(k)

Lemma: Let F be a FVS of graph G. Then,

$$\deg(V\setminus F) \le \deg(F) + 2(|V\setminus F|-1), \quad \text{where } \deg(S) := \sum_{v\in S}\deg(v)$$

Proof:

- G-F is f … G-F.
  Each c …
- Every e … and deg(V\F)

So if success prob ≤ 1/2.99, then must have $m \le 20n \le 80k \Rightarrow m \le O(k)$

If $n \ge 4k$ and F has size k:

$$\sum_{v\notin F}(\deg(v)-3) = \deg(V\setminus F) - 3|V\setminus F|$$
$$\le \deg(F) + 2(|V\setminus F|-1) - 3|V\setminus F|$$
$$\le \deg(F) - 3|V\setminus F|$$

$n \ge 4k$: $|V\setminus F| \ge 3k = 3|F|$

$$\le \deg(F) - 3|F| = \sum_{v\in F}(\deg(v)-3)$$

choose $v \in F$ w.p. $\ge 1/2$

If $m \ge 20n$ and F has size k:

$$\underbrace{\deg(V)}_{2m} - \deg(F) \le \deg(F) + 2(\underbrace{|V\setminus F|}_{n-k}-1)$$

$$\deg(F) \ge \frac{1}{2}(2m - 2(n-k-1)) \ge m-n+k$$

$$\sum_{v\in F}(\deg(v)-3) = \deg(v) - 3|F| \ge m - 4n \ge 0.8m$$

$$\sum_{v\in V}(\deg(v)-3) \le \deg(V) = 2m$$

$\frac{0.8m}{2m} \Rightarrow$ success w.p. ≥ 0.4

Iterative Compression

Original problem: given graph G, find FVS size k, or determine none exist.

Iterative Compression

Original problem: given graph G, find FVS size k, or determine none exist.

Iterative Compression reduces to: given graph G and FVS size (k+1), find FVS size k or determine none exist.

Iterative Compression

Original problem: given graph G, find FVS size k, or determine none exist.

Iterative Compression reduces to: given graph G and FVS size (k+1), find FVS size k or determine none exist.

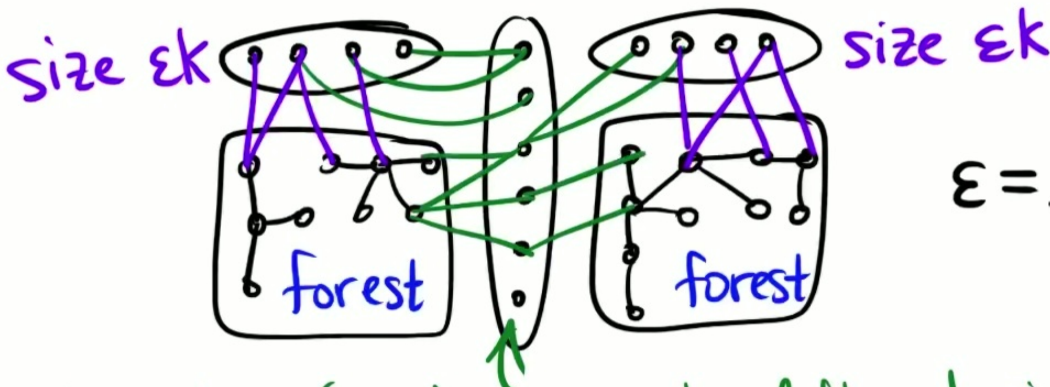
- Get size (k+1) FVS as input for free

Sparse case

Lemma: Given a graph with $m \leq 100k$, and given a FVS of size $k+1$,

Sparse case

Lemma: Given a graph with $m \leq 100k$, and given a FVS of size $k+1$, we can decompose the graph into:

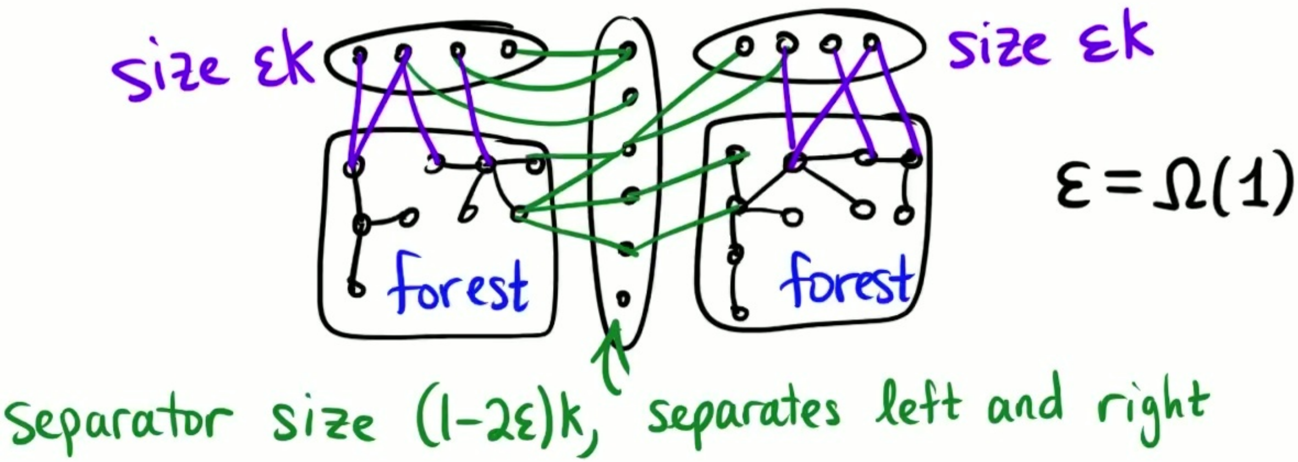


$\varepsilon = \Omega(1)$

separator size $(1-2\varepsilon)k$, separates left and right

Sparse case

Lemma: Given a graph with $m \leq 100k$, and given a FVS of size $k+1$, we can decompose the graph into:

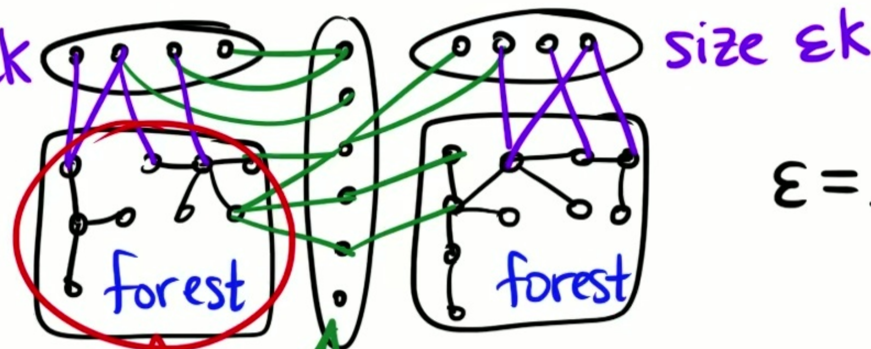


Claim: a graph with this decomposition has treewidth $(1-\Omega(1))k$

Proof:

Sparse case

Lemma: Given a graph with $m \leq 100k$, and given a FVS of size k+1, we can decompose the graph into:



$\varepsilon = \Omega(1)$

Separator size $(1-2\varepsilon)k$, separates left and right

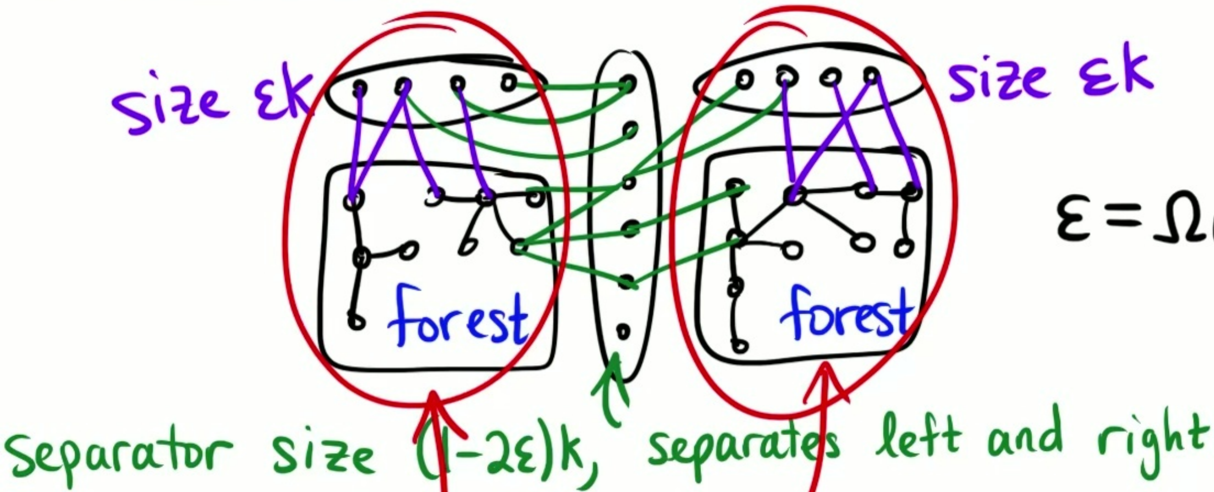
Claim: a graph with this decomposition has treewidth (1-Ω(1))k

Proof:

tw( ) = 1 (forest)

Sparse case

Lemma: Given a graph with $m \leq 100k$, and given a FVS of size $k+1$, we can decompose the graph into:



$\varepsilon = \Omega(1)$

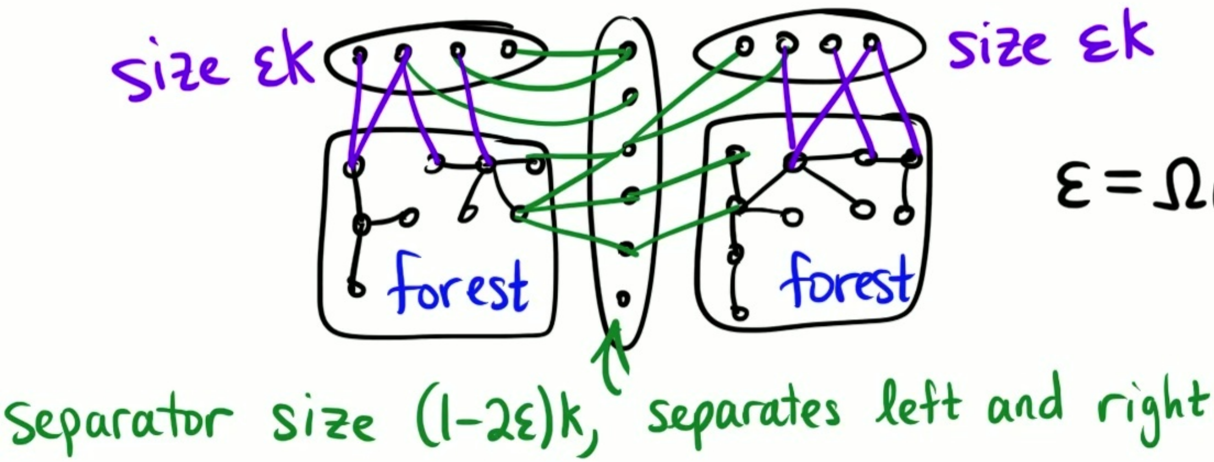
$\mathrm{tw}(\quad) \leq 1 + \varepsilon k$ (forest + εk vertices)

$\mathrm{tw}(G) \leq 1 + \varepsilon k + (1-2\varepsilon)k$ (add last $(1-2\varepsilon)k$ vertices)

$= (1 - \Omega(1))k$

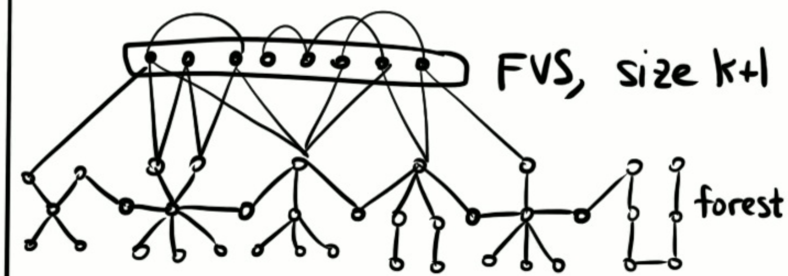
Sparse case

Lemma: Given a graph with m ≤ 100k, and given a FVS of size k+1, we can decompose the graph into:



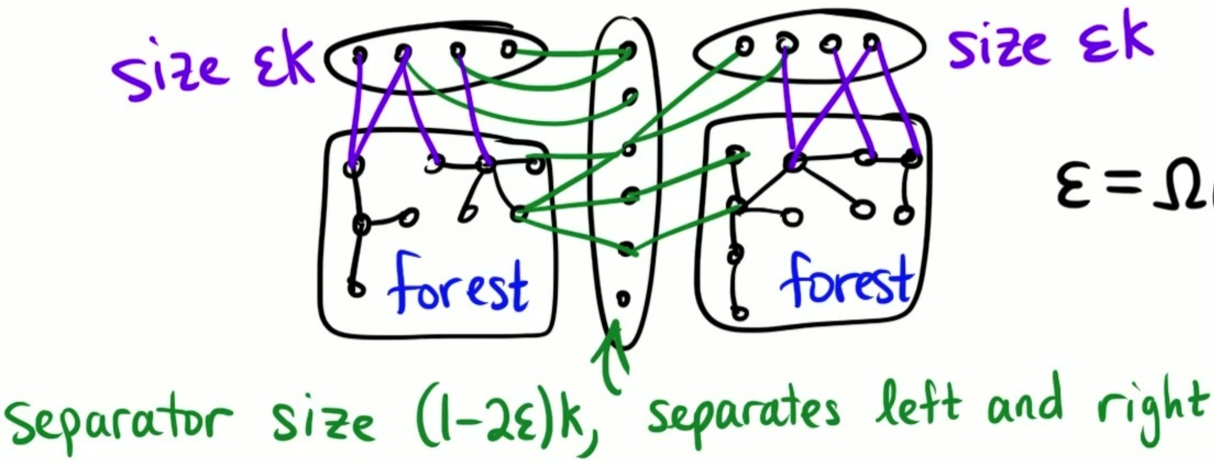
$\varepsilon = \Omega(1)$

Proof of Lemma:



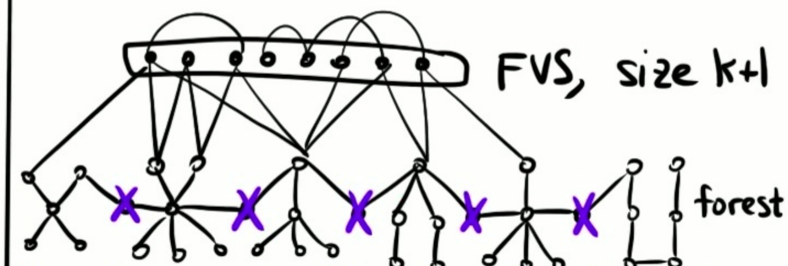
Sparse case

Lemma: Given a graph with m ≤ 100k, and given a FVS of size k+1, we can decompose the graph into:



$\varepsilon = \Omega(1)$

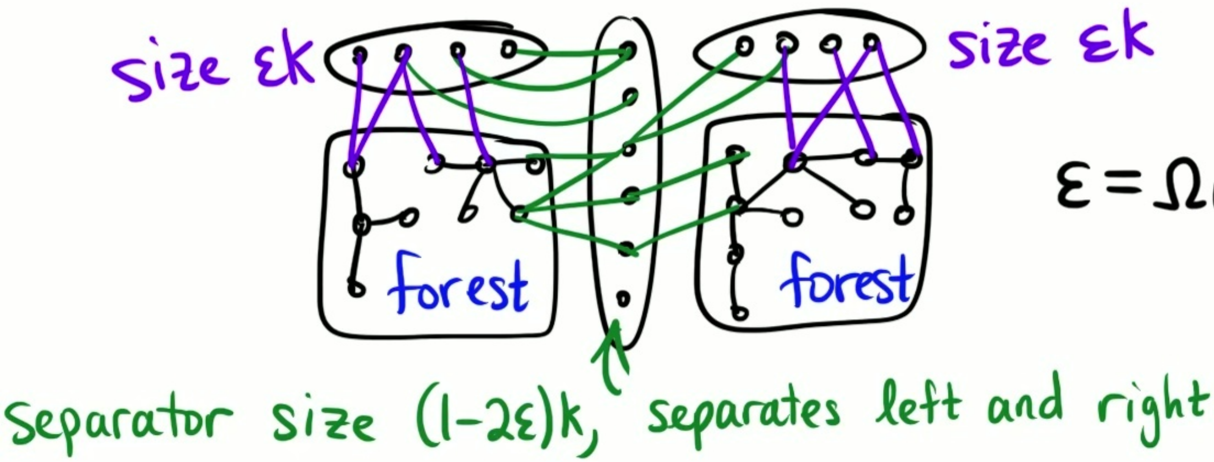
Proof of Lemma:



Step 1: remove t=o(k) vertices from forest s.t. each remaining component is "small". These are added to separator

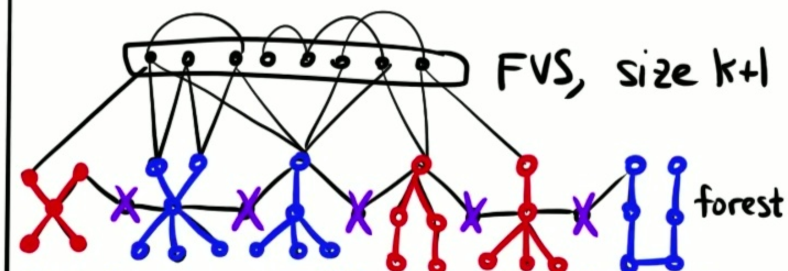
Sparse case

Lemma: Given a graph with $m \leq 100k$, and given a FVS of size $k+1$, we can decompose the graph into:



$\varepsilon = \Omega(1)$

Proof of Lemma:

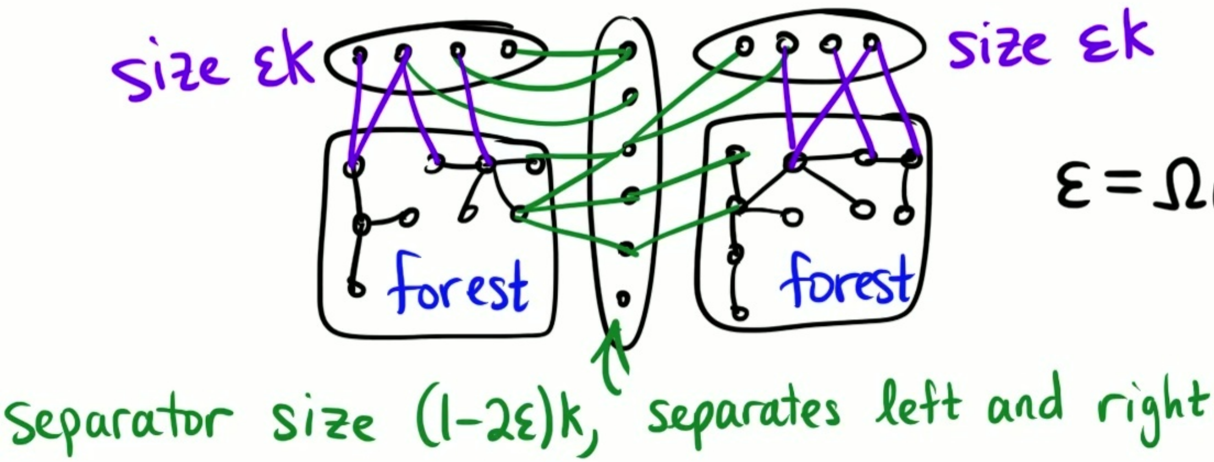


Step 1: remove $t=o(k)$ vertices from forest s.t. each remaining component is "small". These are added to separator

Step 2: for each component, color red/blue w.p. 1/2 each

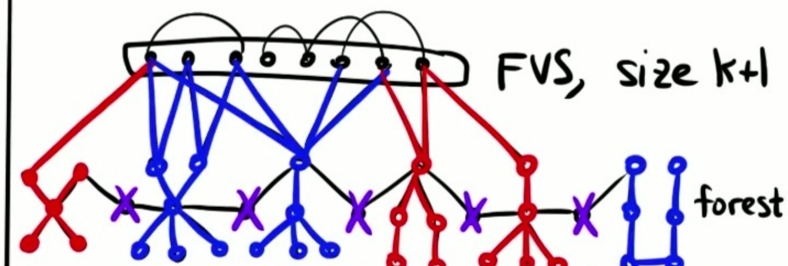
Sparse case

Lemma: Given a graph with m ≤ 100k, and given a FVS of size k+1, we can decompose the graph into:



$\varepsilon = \Omega(1)$

Proof of Lemma:

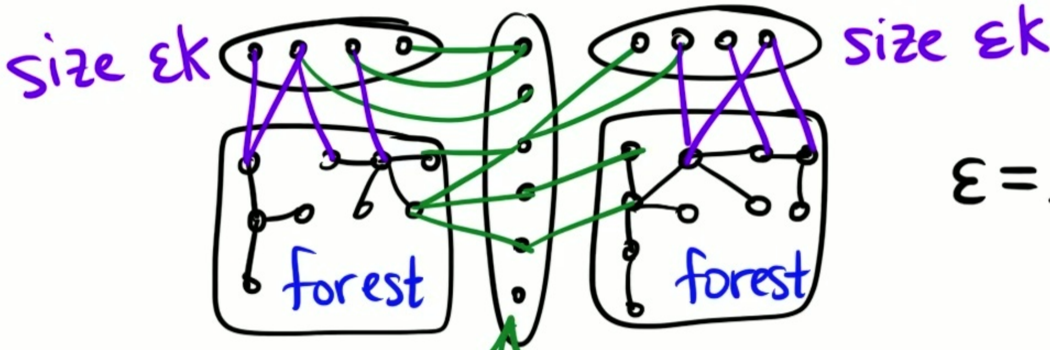


Step 1: remove t=o(k) vertices from forest s.t. each remaining component is "small". These are added to separator

Step 2: for each component, color red/blue w.p. 1/2 each

Sparse case

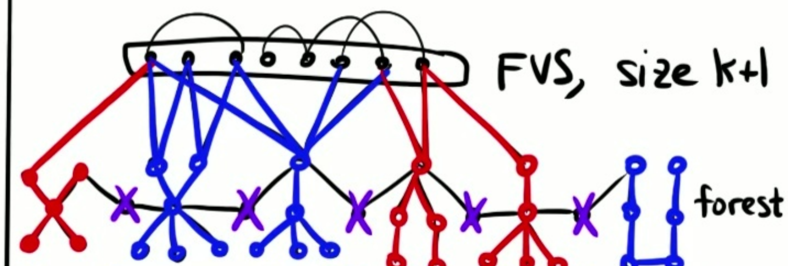
Lemma: Given a graph with m ≤ 100k, and given a FVS of size k+1, we can decompose the graph into:



$\varepsilon = \Omega(1)$

separator size $(1-2\varepsilon)k$, separates left and right

Proof of Lemma:



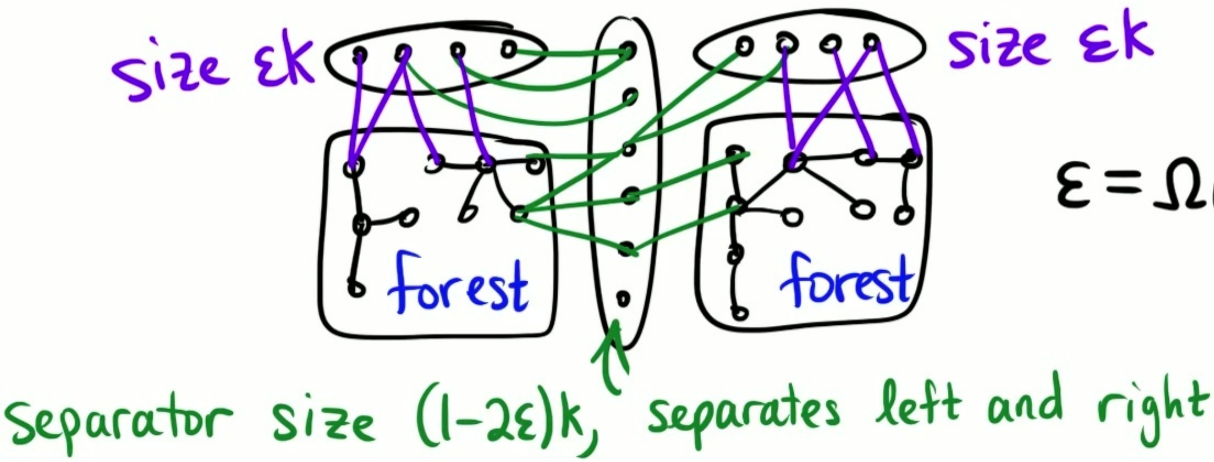
Step 1: remove t=o(k) vertices from forest s.t. each remaining component is "small". These are added to separator

Step 2: for each component, color red/blue w.p. 1/2 each

Step 3: Color each edge between FVS vertices red/blue w.p. 1/2 each

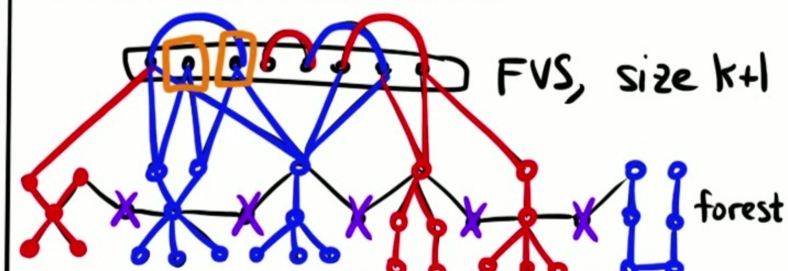
Sparse case

Lemma: Given a graph with $m \le 100k$, and given a FVS of size $k+1$, we can decompose the graph into:



$\varepsilon = \Omega(1)$

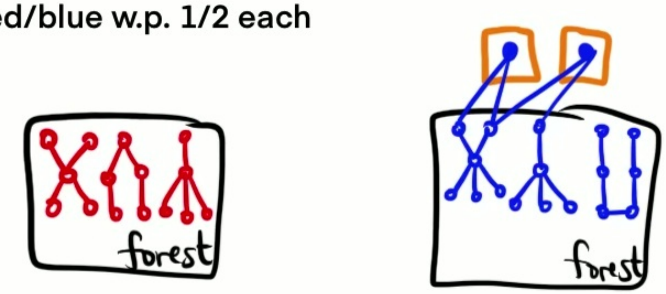
Proof of Lemma:



Step 1: remove $t=o(k)$ vertices from forest s.t. each remaining component is "small". These are added to separator

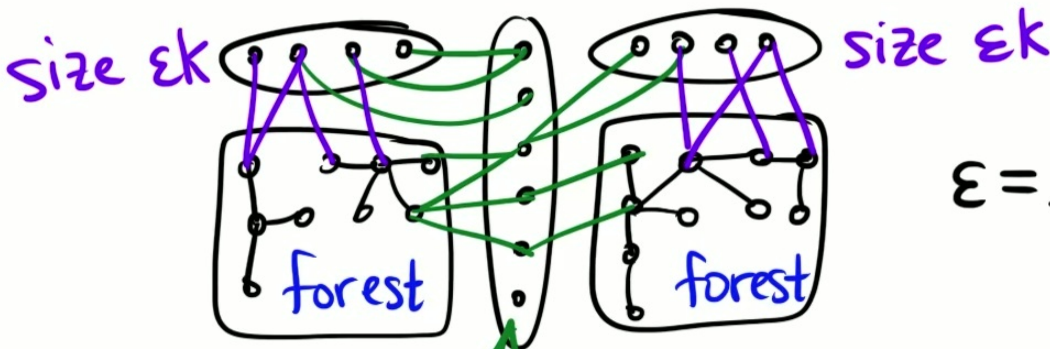
Step 2: for each component, color red/blue w.p. 1/2 each

Step 3: Color each edge between FVS vertices red/blue w.p. 1/2 each



Sparse case

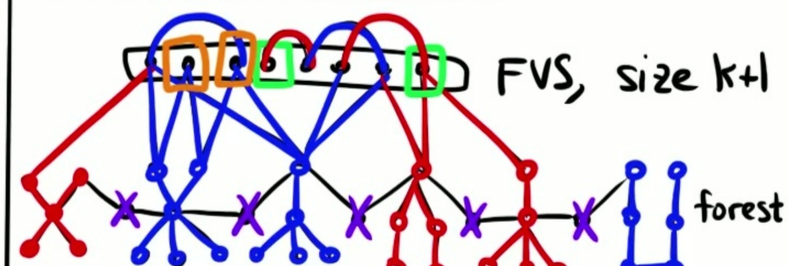
Lemma: Given a graph with $m \le 100k$, and given a FVS of size k+1, we can decompose the graph into:



$\varepsilon = \Omega(1)$

separator size $(1-2\varepsilon)k$, separates left and right

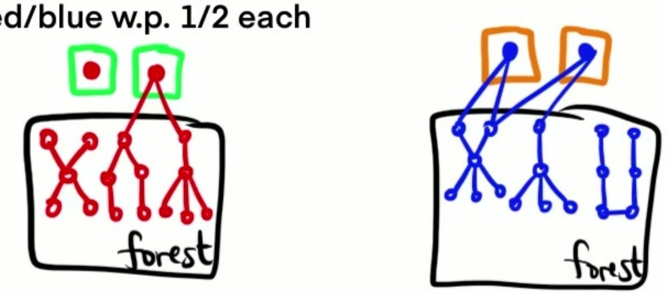
Proof of Lemma:



Step 1: remove $t = o(k)$ vertices from forest s.t. each remaining component is "small". These are added to separator

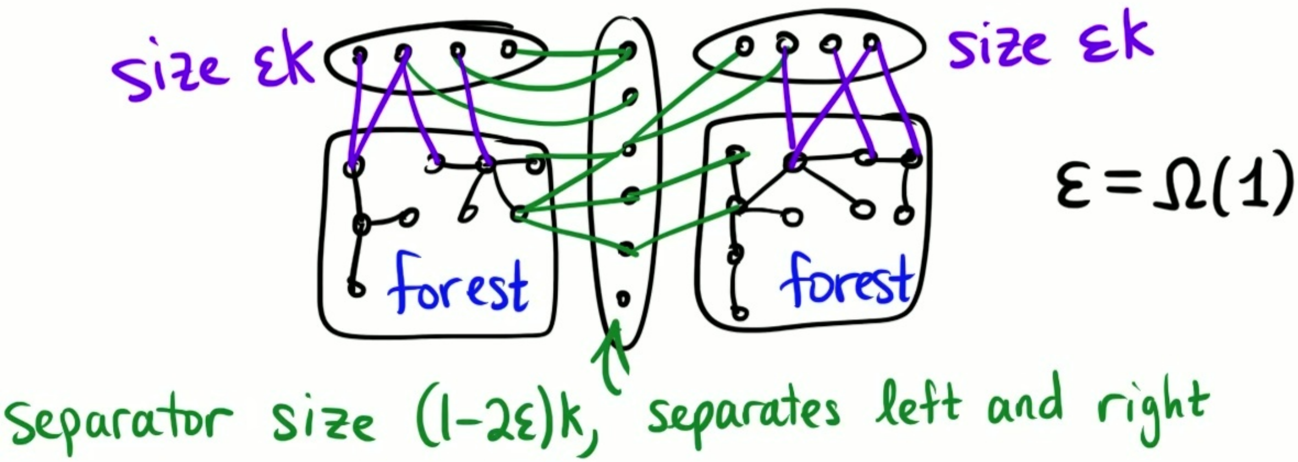
Step 2: for each component, color red/blue w.p. 1/2 each

Step 3: Color each edge between FVS vertices red/blue w.p. 1/2 each

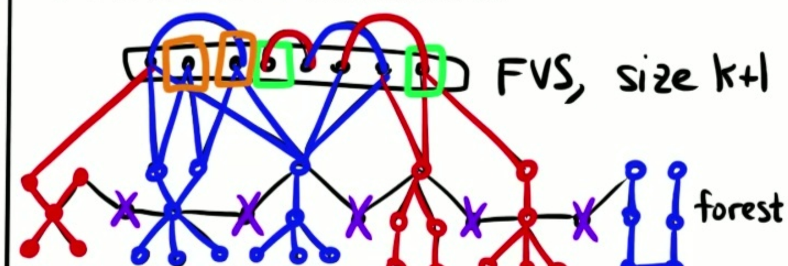


Sparse case

Lemma: Given a graph with m ≤ 100k, and given a FVS of size k+1, we can decompose the graph into:



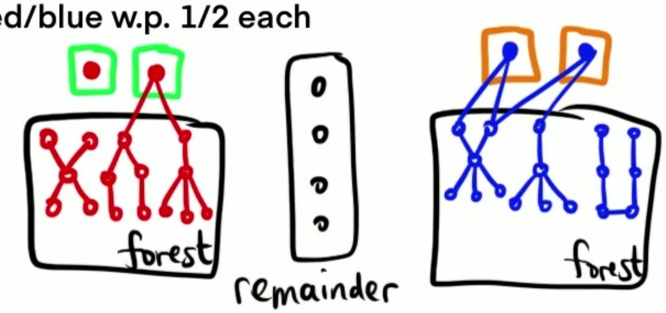
Proof of Lemma:



Step 1: remove t=o(k) vertices from forest s.t. each remaining component is "small". These are added to separator

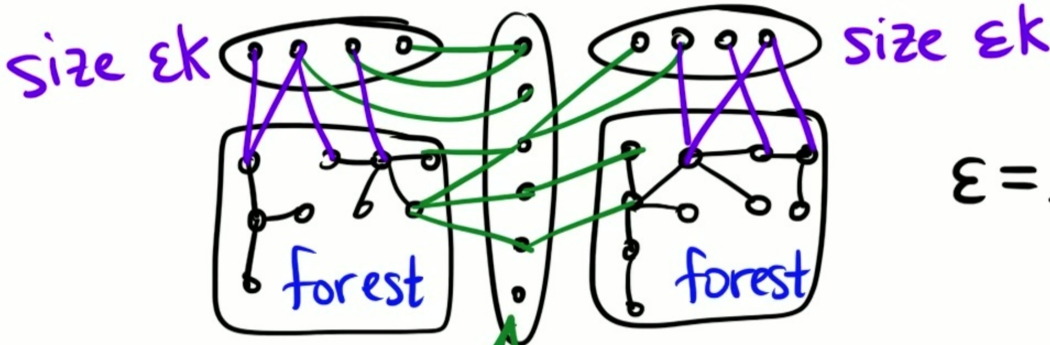
Step 2: for each component, color red/blue w.p. 1/2 each

Step 3: Color each edge between FVS vertices red/blue w.p. 1/2 each



Sparse case

Lemma: Given a graph with $m \leq 100k$, and given a FVS of size $k+1$, we can decompose the graph into:

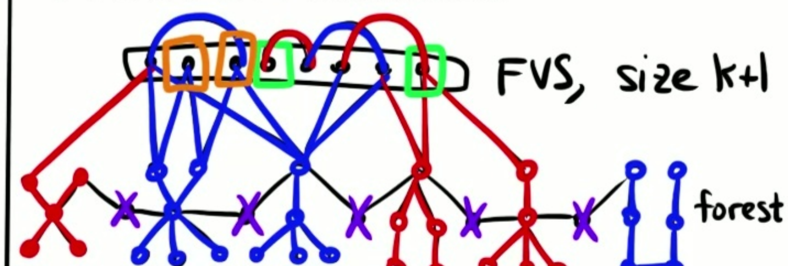


$\varepsilon = \Omega(1)$

separator size $(1-2\varepsilon)k$, separates left and right

$\Pr[\text{all incident edges red}] \geq 2^{-\deg(v)}$

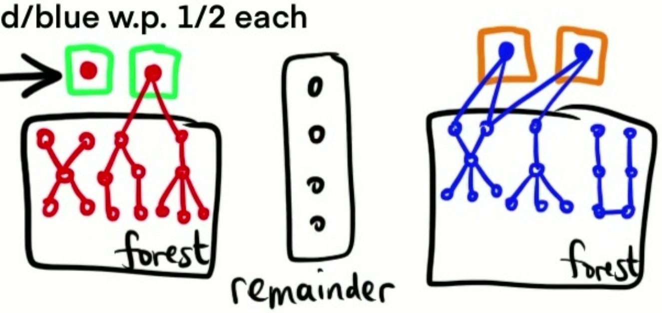
Proof of Lemma:



Step 1: remove t=o(k) vertices from forest s.t. each remaining component is "small". These are added to separator

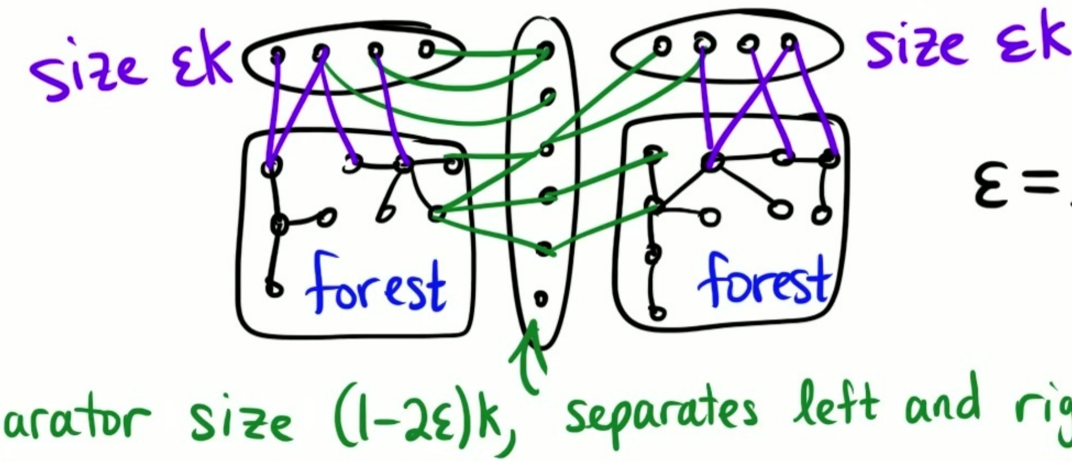
Step 2: for each component, color red/blue w.p. 1/2 each

Step 3: Color each edge between FVS vertices red/blue w.p. 1/2 each



Sparse case

Lemma: Given a graph with m ≤ 100k, and given a FVS of size k+1, we can decompose the graph into:

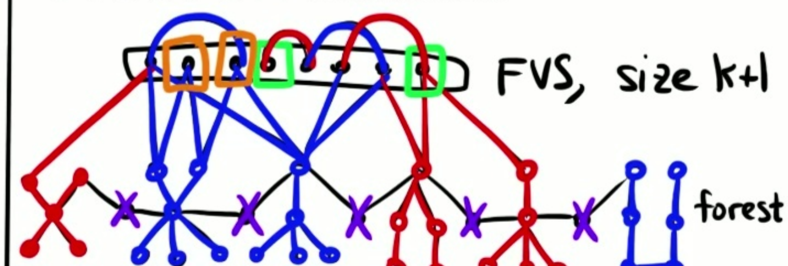


$\varepsilon = \Omega(1)$

$$\Pr[\text{all incident edges red}] \geq 2^{-\deg(v)}$$

$$\mathbb{E}[\#\text{ such } v] \geq \sum_{v \in FVS} 2^{-\deg(v)} \overset{\text{Jensen}}{\geq} |FVS| \cdot 2^{-(\text{avg deg})}$$

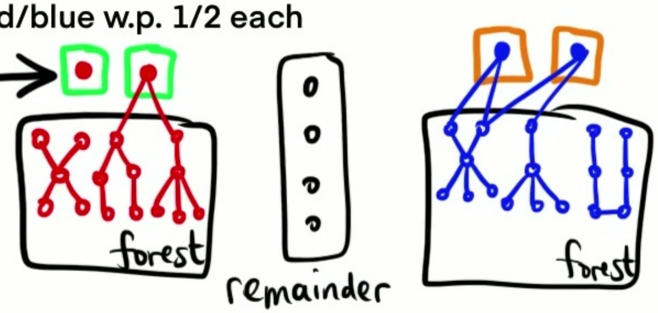
Proof of Lemma:



Step 1: remove t=o(k) vertices from forest s.t. each remaining component is "small". These are added to separator

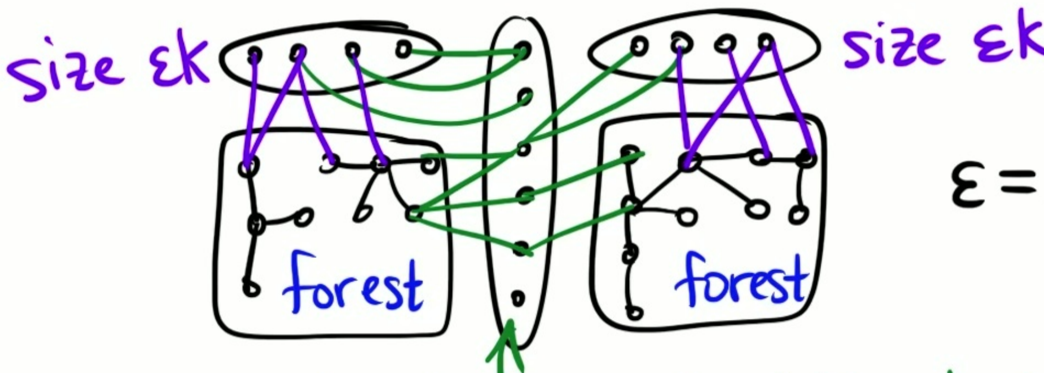
Step 2: for each component, color red/blue w.p. 1/2 each

Step 3: Color each edge between FVS vertices red/blue w.p. 1/2 each



Sparse case

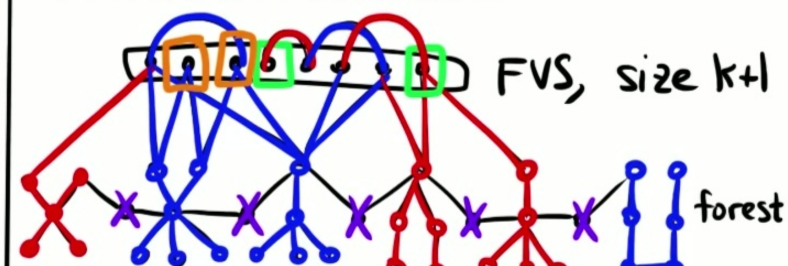
Lemma: Given a graph with m ≤ 100k, and given a FVS of size k+1, we can decompose the graph into:



$\varepsilon = \Omega(1)$

separator size $(1-2\varepsilon)k$, separates left and right

Proof of Lemma:



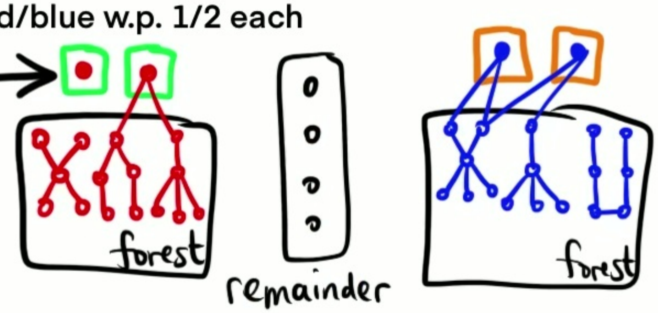
Step 1: remove t=o(k) vertices from forest s.t. each remaining component is "small". These are added to separator

Step 2: for each component, color red/blue w.p. 1/2 each

Step 3: Color each edge between FVS vertices red/blue w.p. 1/2 each

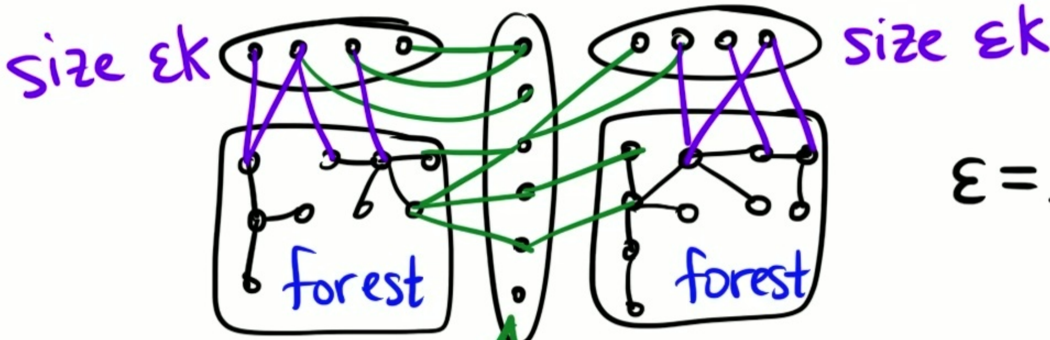
$\Pr[\text{all incident edges red}] \geq 2^{-\deg(v)}$

$\mathbb{E}[\#\text{ such } v] \geq \sum_{v \in FVS} 2^{-\deg(v)} \overset{\text{Jensen}}{\geq} \underbrace{|FVS|}_{k+1} \cdot 2^{-(\overbrace{\text{avg deg}}^{\leq \frac{2m}{k+1} \leq 200})} \geq 2^{-200}k.$



Sparse case

Lemma: Given a graph with $m \le 100k$, and given a FVS of size $k+1$, we can decompose the graph into:



$\varepsilon = \Omega(1)$

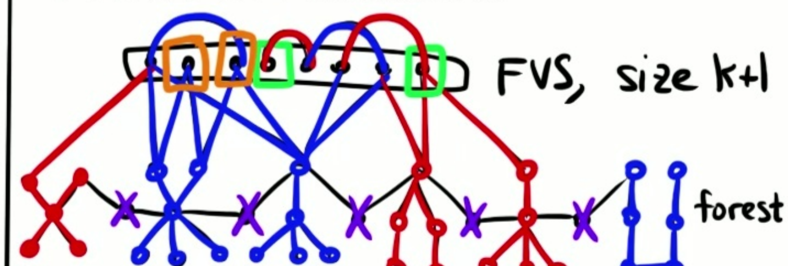
Separator size $(1-2\varepsilon)k$, separates left and right

$\Pr[\text{all incident edges red}] \ge 2^{-\deg(v)}$

$\mathbb{E}[\#\text{ such } v] \ge \sum_{v \in FVS} 2^{-\deg(v)} \overset{\text{Jensen}}{\ge} \underbrace{|FVS|}_{k+1} \cdot 2^{-\overbrace{(\text{avg deg})}^{\le \frac{2m}{k+1} \le 200}} \ge 2^{-200} k.$

Chernoff bound (since each X is "small"): $\#$ ☐ $\approx \#$ ☐ $\approx \varepsilon k$ for some $\varepsilon \ge 2^{-200}$

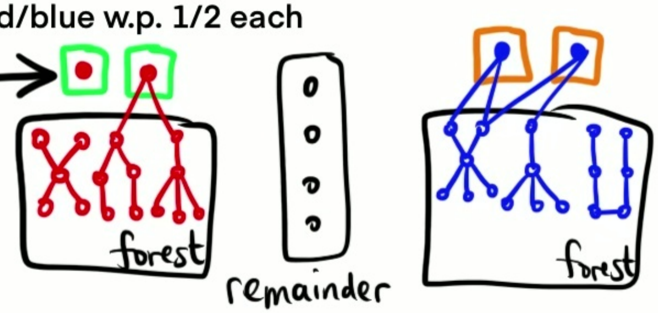
Proof of Lemma:



Step 1: remove $t=o(k)$ vertices from forest s.t. each remaining component is "small". These are added to separator

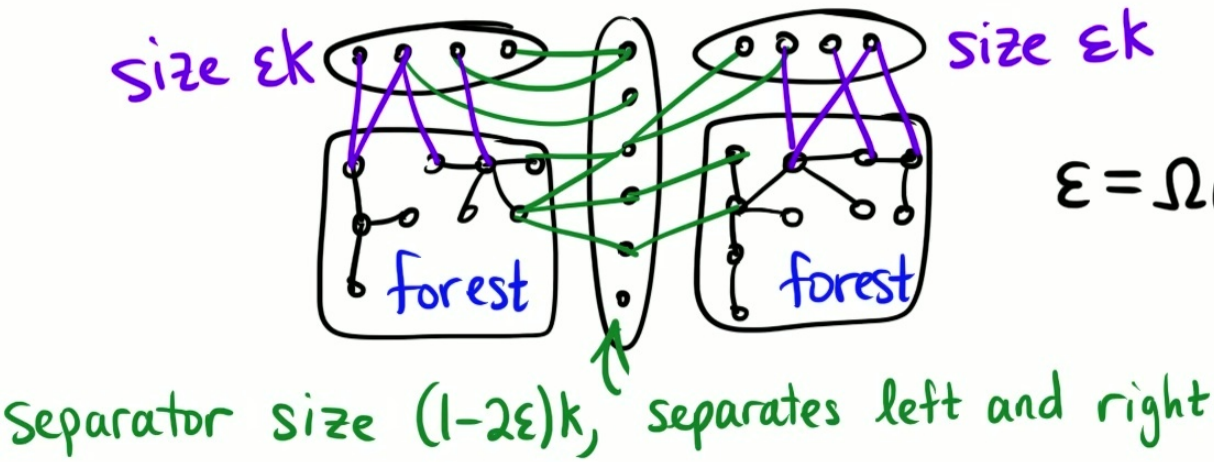
Step 2: for each component, color red/blue w.p. 1/2 each

Step 3: Color each edge between FVS vertices red/blue w.p. 1/2 each



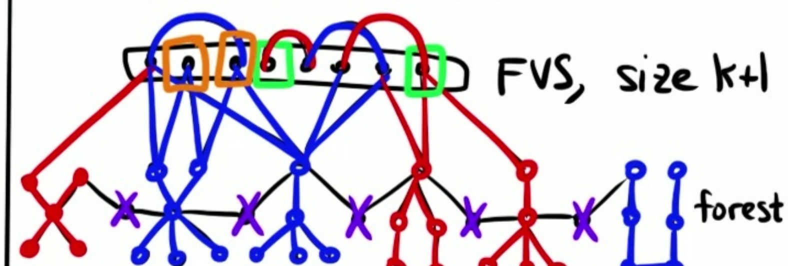
Sparse case

Lemma: Given a graph with $m \le 100k$, and given a FVS of size $k+1$, we can decompose the graph into:



$\varepsilon = \Omega(1)$

Proof of Lemma:



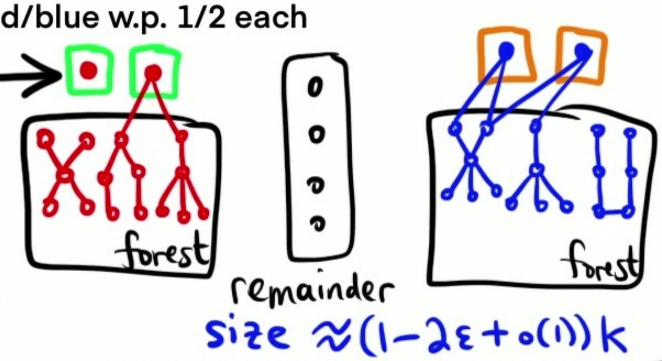
Step 1: remove $t=o(k)$ vertices from forest s.t. each remaining component is "small". These are added to separator

Step 2: for each component, color red/blue w.p. 1/2 each

Step 3: Color each edge between FVS vertices red/blue w.p. 1/2 each

$$\Pr[\text{all incident edges red}] \ge 2^{-\deg(v)}$$

$$\mathbb{E}[\#\text{ such } v] \ge \sum_{v \in FVS} 2^{-\deg(v)} \underset{\text{Jensen}}{\ge} \overset{k+1}{\overbrace{|FVS|}} \cdot 2^{-(\overbrace{\text{avg deg}}^{\le \frac{2m}{k+1} \le 200})} \ge 2^{-200} k.$$



Chernoff bound (since each is "small"): # ≈ # ≈ εk for some $\varepsilon \ge 2^{-200}$

Speedup: O*(2.7^k) time

- Tighten (deg(v)-3) analysis and open 3^tw algorithm [CNP+11]
- [CNP+11] actually solves a counting problem
  - special arithmetic structure: speed up via fast matrix multiplication

Speedup: O*(2.7^k) time

- Tighten (deg(v)-3) analysis and open 3^tw algorithm [CNP+11]
- [CNP+11] actually solves a counting problem
  - special arithmetic structure: speed up via fast matrix multiplication

Open problems

- Our main conceptual message: 3^k can be broken (randomized)
  - Faster deterministic algorithm? [BBG'00] is inherently randomized
- 2^k possible?
- SETH lower bound? No 1.00001^k lower bound known!