Parameterized Algorithms

Lecture 2: Introduction cont...

May 15, 2020

Max-Planck Institute for Informatics, Germany.

Recall....

Vertex Cover

- **Input :** Graph G on n vertices and integer k
- **Parameter :** k
- **Output :** $S \subseteq V(G)$ such that $G - S$ is edgeless and $|S| \leq k$.

Iterative Compression

Input

A graph $G = (V,E)$ with n vertices, m edges, and k.

Is there a vertex cover of size at most k?

Question

Vertex Cover

Input A graph $G = (V,E)$ with n vertices, m edges, and k.

Is there a vertex cover of size at most k? **Question**

A vertex cover of size $(k+1)$.

Vertex Cover

Input

A graph G with a vertex cover of size k+1.

Is there a vertex cover of size at most k?

Question

A vertex cover of size (k+1).

Vertex Cover

Input

A graph G with a vertex cover of size k+1.

Is there a vertex cover of size at most k?

Question

A vertex cover of size (k+1).

Vertex Cover

Input

A graph G with a vertex cover of size k+1.

Is there a vertex cover of size at most k?

Question

A vertex cover of size (k+1).

Let us "guess" how a vertex cover of size at most k interacts with this one.

VERTEX COVER

Input A graph G with a vertex cover of size k+1.

Is there a vertex cover of size at most k? **Question**

A vertex cover of size (k+1).

Let us "guess" how a vertex cover of size at most k interacts with this one.

Vertex Cover

Input

A graph G with a vertex cover of size k+1.

Is there a vertex cover of size at most k?

Question

A vertex cover of size (k+1).

Let us "guess" how a vertex cover of size at most k interacts with this one.

VERTEX COVER

Input A graph G with a vertex cover of size k+1.

Question Is there a vertex cover of size at most k?

A vertex cover of size (k+1).

Vertex Cover

Input

A graph G with a vertex cover of size k+1.

Is there a vertex cover of size at most k?

Question

A vertex cover of size (k+1).

Can there be an edge between two red vertices?

VERTEX COVER

Input

A graph G with a vertex cover of size k+1.

Is there a vertex cover of size at most k?

Question

A vertex cover of size (k+1).

Vertex Cover

Input

A graph G with a vertex cover of size k+1.

Is there a vertex cover of size at most k?

Question

A vertex cover of size (k+1).

Now we need to make up for the work that the red vertices were doing.

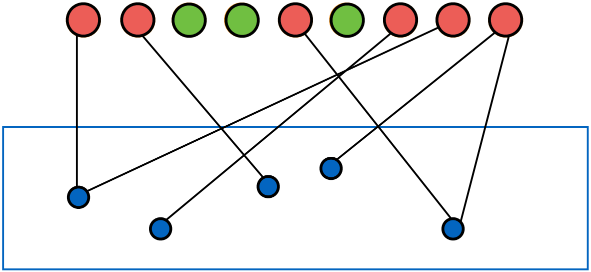
VERTEX COVER

Input

A graph G with a vertex cover of size k+1.

Is there a vertex cover of size at most k?

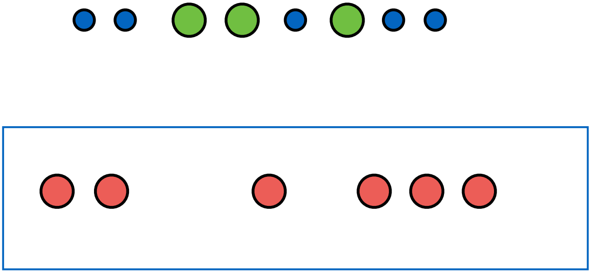
Question



Vertex Cover

Input A graph G with a vertex cover of size k+1.

Question Is there a vertex cover of size at most k?



Vertex Cover

Input

A graph $G = (V,E)$ with n vertices, m edges, and k.

Is there a vertex cover of size at most k?

Question

Vertex Cover

Input A graph G = (V,E) with n vertices, m edges, and k.

Is there a vertex cover of size at most k? **Question**

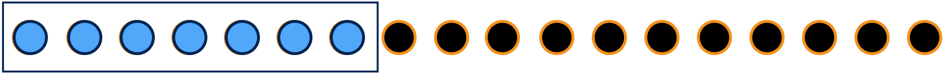
Vertex Cover

Input

A graph $G = (V,E)$ with n vertices, m edges, and k.

Is there a vertex cover of size at most k?

Question



The first $k+2$ vertices in G.

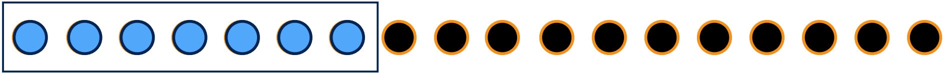
Vertex Cover

Input

A graph $G = (V,E)$ with n vertices, m edges, and k.

Is there a vertex cover of size at most k?

Question



The first $k+2$ vertices in G. It has an easy vertex cover of size $k+1$.

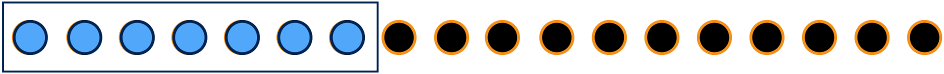
Vertex Cover

Input

A graph G = (V,E) with n vertices, m edges, and k.

Is there a vertex cover of size at most k?

Question

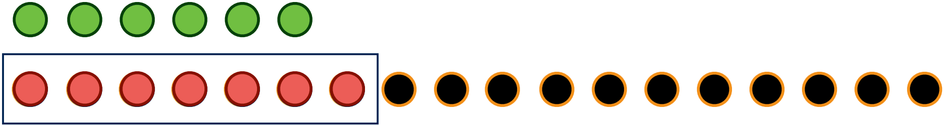


The first k+2 vertices in G. It has an easy vertex cover of size k+1.

Vertex Cover

Input A graph G = (V,E) with n vertices, m edges, and k.

Is there a vertex cover of size at most k? **Question**



The first k+2 vertices in G. It has an easy vertex cover of size k+1.

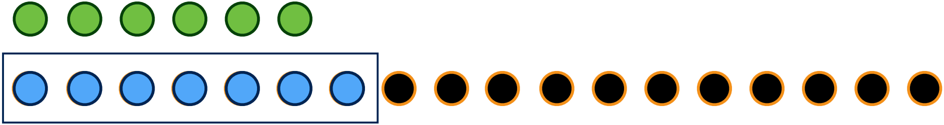
Vertex Cover

Input

A graph G = (V,E) with n vertices, m edges, and k.

Is there a vertex cover of size at most k?

Question

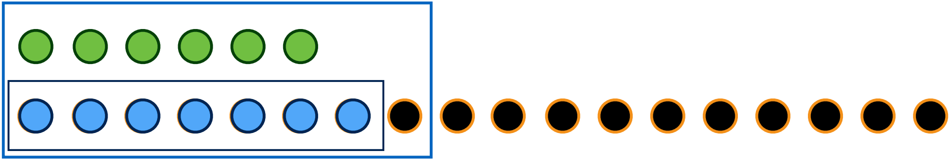


The first k+2 vertices in G. It has an easy vertex cover of size k+1.

Vertex Cover

Input A graph G = (V,E) with n vertices, m edges, and k.

Is there a vertex cover of size at most k? **Question**

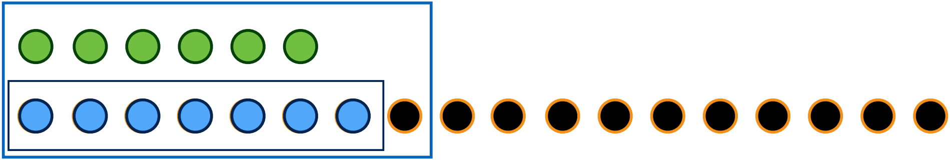


The first k+2 vertices in G. It has an easy vertex cover of size k+1.

Vertex Cover

Input A graph G = (V,E) with n vertices, m edges, and k.

Is there a vertex cover of size at most k? **Question**



The first k+2 vertices in G. It has an easy vertex cover of size k+1.

Compress again… rinse, repeat.

Vertex Cover

Input

A graph $G = (V,E)$ with n vertices, m edges, and k.

Is there a subset of vertices S of size at most k that intersects all the edges?

Question

We have a $O^*(2^k)$ algorithm for Vertex Cover.

Vertex Cover

Recall....

Feedback Vertex Set

- **Input :** Graph G on n vertices and integer k
- **Parameter :** k
- **Output :** $S \subseteq V(G)$ such that $G - S$ is acyclic and $|S| \leq k$.

Input

A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic?

Question

A Feedback Vertex Set of size (k+1).

Let us "guess" how a FVS of size at most k interacts with this one.

Feedback Vertex Set

Input A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic? **Question**

Feedback Vertex Set

Input A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic? **Question**

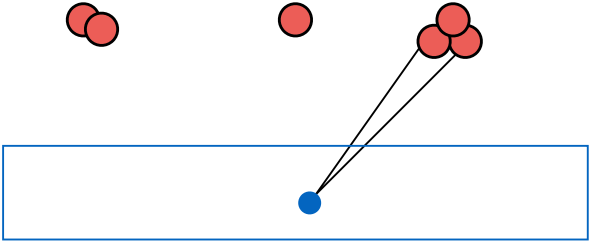
Feedback Vertex Set

Input

A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic?

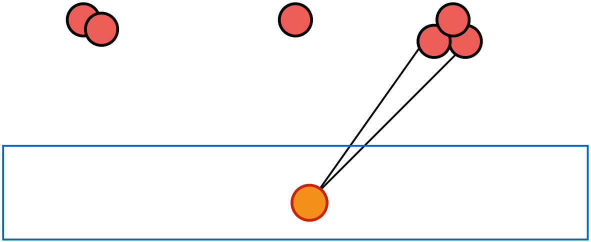
Question



FEEDBACK VERTEX SET

Input A graph on n vertices, m edges and an integer k.

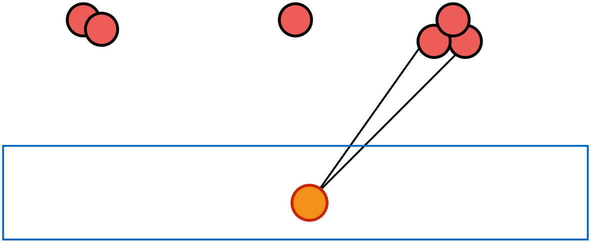
Is there a subset of at most k vertices whose removal makes the graph acyclic? **Question**



Feedback Vertex Set

Input A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic? Question



A vertex with two neighbors in the same component is forced.

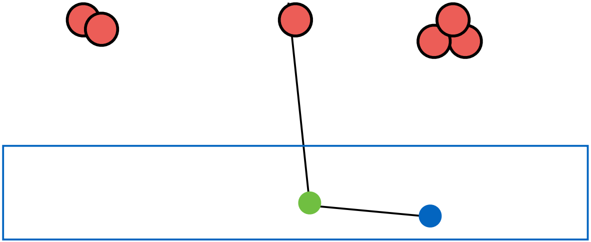
FEEDBACK VERTEX SET

Input

A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic?

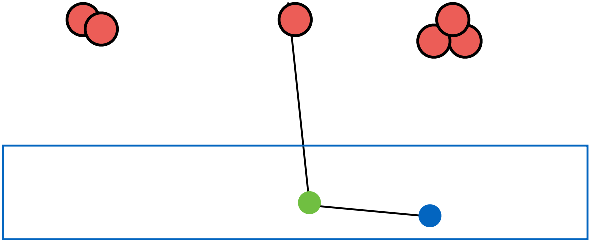
Question



FEEDBACK VERTEX SET

Input A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic? Question



A leaf that has exactly one neighbor above can be preprocessed.

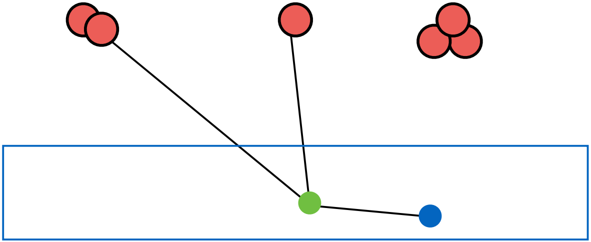
FEEDBACK VERTEX SET

Input

A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic?

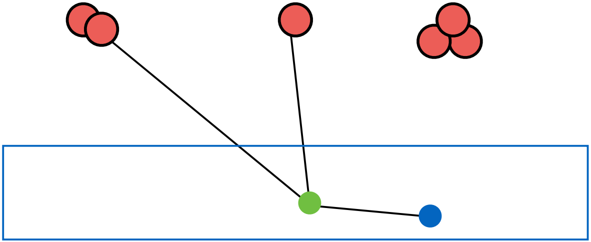
Question



FEEDBACK VERTEX SET

Input A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic? **Question**

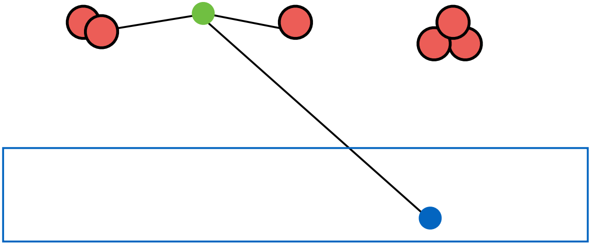


...a leaf with at least two neighbors in different components.

Feedback Vertex Set

Input A graph on n vertices, m edges and an integer k.

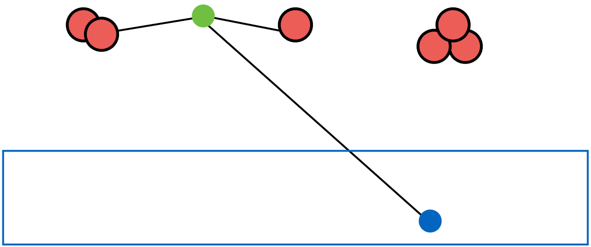
Is there a subset of at most k vertices whose removal makes the graph acyclic? **Question**



FEEDBACK VERTEX SET

Input A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic? **Question**



The leaf merges two components when we don't include it.

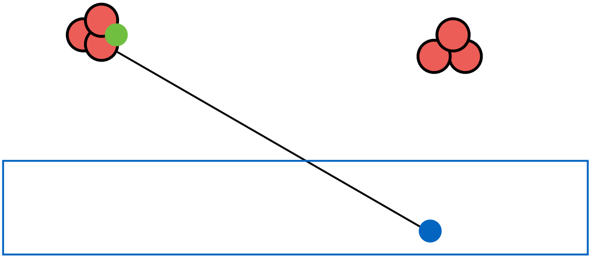
FEEDBACK VERTEX SET

Input

A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic?

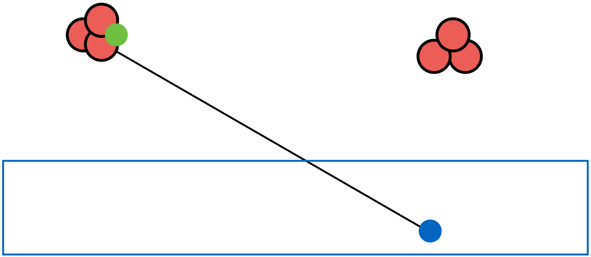
Question



FEEDBACK VERTEX SET

Input A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic? Question



The number of components "on top" decreases.

Feedback Vertex Set

Input A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic? **Question**

Start with a leaf.

Two neighbors in one component: forced.
At most one neighbor above: preprocess.
At least two neighbors, all in different components above: branch.

Feedback Vertex Set

Input

A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic?

Question

FEEDBACK VERTEX SET

Input A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic? **Question**

Let t denote the number of components among the red vertices. Let w = (k+t).

Feedback Vertex Set

Input

A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic?

Question

Let t denote the number of components among the red vertices. Let w = (k+t).

Include v… k drops by 1
Exclude v… t drops by at least 1

Feedback Vertex Set

Input

A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic?

Question

Let t denote the number of components among the red vertices. Let w = (k+t).

Include v… k drops by 1
Exclude v… t drops by at least 1

Either way, w drops by at least one.

FEEDBACK VERTEX SET

Input

A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic?

Question

Either way, w drops by at least one.

Feedback Vertex Set

Input

A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic?

Question

Either way, w drops by at least one.

Running time: $2^w = 2^{(k+t)}$

Feedback Vertex Set

Input

A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic?

Question

Either way, w drops by at least one.

Running time: $2^w = 2^{(k+t)} \leq 2^{k+k} \leq 4^k$

Feedback Vertex Set

Input

A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic?

Question

Either way, w drops by at least one.

Running time: $2^w = 2^{(k+t)} \leq 2^{k+k} \leq 4^k$

Overall Running Time…

Feedback Vertex Set

Input A graph on n vertices, m edges and an integer k.

Is there a subset of at most k vertices whose removal makes the graph acyclic? **Question**

Either way, w drops by at least one.

Running time: $2^w = 2^{(k+t)} \leq 2^{k+k} \leq 4^k$

Overall Running Time...

$$\sum_{i=1}^{k} \binom{k}{i} 4^k = 5^k$$

Feedback Vertex Set

Input

A tournament T on n vertices and an integer k.

Is there a subset of k arcs that can be reversed to make the tournament acyclic?

Question

This implies an $O(5^k)$ algorithm for FVS.

Feedback Vertex Set

Bipartite Deletion

Also Known as ODD CYCLE TRANSVERSAL

- **Input :** Graph G on n vertices and integer k
- **Parameter :** k
- **Output :** $S \subseteq V(G)$ such that $G - S$ is Bipartite and $|S| \leq k$.

Bipartite Graphs:

- Are 2-colorable
- Have no odd cycle

BIPARTITE DELETION

Solution based on iterative compression:

- **Step 1:**

  Solve the **annotated problem** for bipartite graphs:

  Given a bipartite graph G, two sets $B, W \subseteq V(G)$, and an integer k, find a set S of at most k vertices such that $G \setminus S$ has a 2-coloring where $B \setminus S$ is black and $W \setminus S$ is white.

BIPARTITE DELETION

Solution based on iterative compression:

- **Step 1:**
  Solve the **annotated problem** for bipartite graphs:

  Given a bipartite graph G, two sets $B, W \subseteq V(G)$, and an integer k, find a set S of at most k vertices such that $G \setminus S$ has a 2-coloring where $B \setminus S$ is black and $W \setminus S$ is white.

- **Step 2:**
  Solve the **compression problem** for general graphs:

  Given a graph G, an integer k, and a set S' of $k+1$ vertices such that $G \setminus S'$ is bipartite, find a set S of k vertices such that $G \setminus S$ is bipartite.

Bipartite Deletion

Solution based on iterative compression:

- **Step 1:**
  Solve the **annotated problem** for bipartite graphs:

  Given a bipartite graph G, two sets $B, W \subseteq V(G)$, and an integer k, find a set S of at most k vertices such that $G \setminus S$ has a 2-coloring where $B \setminus S$ is black and $W \setminus S$ is white.

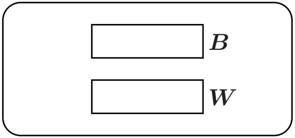
- **Step 2:**
  Solve the **compression problem** for general graphs:

  Given a graph G, an integer k, and a set S' of $k+1$ vertices such that $G \setminus S'$ is bipartite, find a set S of k vertices such that $G \setminus S$ is bipartite.

- **Step 3:**
  Apply the magic of iterative compression...

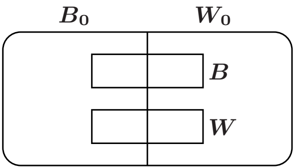
Step 1: The annotated problem

Given a bipartite graph G, two sets $B, W \subseteq V(G)$, and an integer k, find a set S of at most k vertices such that $G \setminus S$ has a 2-coloring where $B \setminus S$ is black and $W \setminus S$ is white.



Step 1: The annotated problem

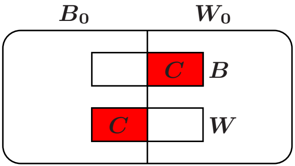
Given a bipartite graph G, two sets $B, W \subseteq V(G)$, and an integer k, find a set S of at most k vertices such that $G \setminus S$ has a 2-coloring where $B \setminus S$ is black and $W \setminus S$ is white.



Find an arbitrary 2-coloring (B_0, W_0) of G.

Step 1: The annotated problem

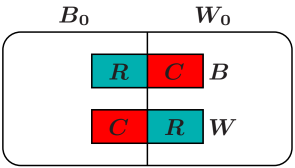
Given a bipartite graph G, two sets $B, W \subseteq V(G)$, and an integer k, find a set S of at most k vertices such that $G \setminus S$ has a 2-coloring where $B \setminus S$ is black and $W \setminus S$ is white.



Find an arbitrary 2-coloring (B_0, W_0) of G.
$C := (B_0 \cap W) \cup (W_0 \cap B)$ should change color, while
$R := (B_0 \cap B) \cup (W_0 \cap W)$ should remain the same color.

Step 1: The annotated problem

Given a bipartite graph G, two sets $B, W \subseteq V(G)$, and an integer k, find a set S of at most k vertices such that $G \setminus S$ has a 2-coloring where $B \setminus S$ is black and $W \setminus S$ is white.



Find an arbitrary 2-coloring (B_0, W_0) of G.
$C := (B_0 \cap W) \cup (W_0 \cap B)$ should change color, while
$R := (B_0 \cap B) \cup (W_0 \cap W)$ should remain the same color.

Lemma: $G \setminus S$ has the required 2-coloring if and only if S separates C and R, i.e., no component of $G \setminus S$ contains vertices from both $C \setminus S$ and $R \setminus S$.

Step 1: The annotated problem

Lemma: $G \setminus S$ has the required 2-coloring if and only if S separates C and R, i.e., no component of $G \setminus S$ contains vertices from both $C \setminus S$ and $R \setminus S$.

Proof:

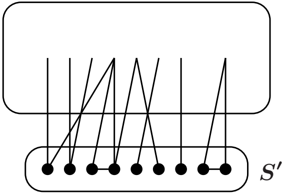
$\Rightarrow$ In a 2-coloring of $G \setminus S$, each vertex either remained the same color or changed color. Adjacent vertices do the same, thus every component either changed or remained.

$\Leftarrow$ Flip the coloring of those components of $G \setminus S$ that contain vertices from $C \setminus S$. No vertex of R is flipped.

Algorithm: Using max-flow min-cut techniques, we can check if there is a set S that separates C and R. It can be done in time $O(k|E(G)|)$ using k iterations of the Ford-Fulkerson algorithm.

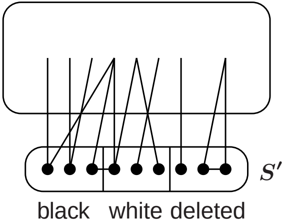
Step 2: The compression problem

Given a graph G, an integer k, and a set S' of $k+1$ vertices such that $G \setminus S'$ is bipartite, find a set S of k vertices such that $G \setminus S$ is bipartite.



Step 2: The compression problem

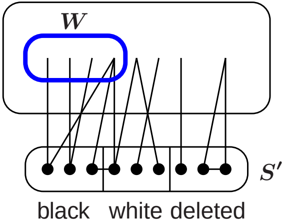
Given a graph G, an integer k, and a set S' of $k+1$ vertices such that $G \setminus S'$ is bipartite, find a set S of k vertices such that $G \setminus S$ is bipartite.



Branch into 3^{k+1} cases: each vertex of S' is either black, white, or deleted.
Trivial check: no edge between two black or two white vertices.

Step 2: The compression problem

Given a graph G, an integer k, and a set S' of $k+1$ vertices such that $G \setminus S'$ is bipartite, find a set S of k vertices such that $G \setminus S$ is bipartite.



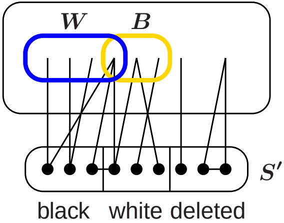
Branch into 3^{k+1} cases: each vertex of S' is either black, white, or deleted.

Trivial check: no edge between two black or two white vertices.

Neighbors of the black vertices in S' should be white and the neighbors of the white vertices in S' should be black.

Step 2: The compression problem

Given a graph G, an integer k, and a set S' of $k+1$ vertices such that $G \setminus S'$ is bipartite, find a set S of k vertices such that $G \setminus S$ is bipartite.



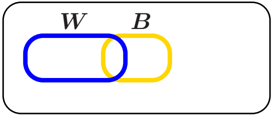
Branch into 3^{k+1} cases: each vertex of S' is either black, white, or deleted.

Trivial check: no edge between two black or two white vertices.

Neighbors of the black vertices in S' should be white and the neighbors of the white vertices in S' should be black.

Step 2: The compression problem

Given a graph G, an integer k, and a set S' of $k+1$ vertices such that $G \setminus S'$ is bipartite, find a set S of k vertices such that $G \setminus S$ is bipartite.



The vertices of S' can be disregarded. Thus we need to solve the annotated problem on the bipartite graph $G \setminus S'$.

Running time: $O(3^k \cdot k|E(G)|)$ time.

Step 3: Iterative compression

How do we get a solution of size $k + 1$?

Step 3: Iterative compression

How do we get a solution of size $k+1$?

We get it for free!

Step 3: Iterative compression

How do we get a solution of size $k+1$?

We get it for free!

Let $V(G) = \{v_1, \ldots, v_n\}$ and let G_i be the graph induced by $\{v_1, \ldots, v_i\}$.

For every i, we find a set S_i of size k such that $G_i \setminus S_i$ is bipartite.

- For G_k, the set $S_k = \{v_1, \ldots, v_k\}$ is a trivial solution.
- If S_{i-1} is known, then $S_{i-1} \cup \{v_i\}$ is a set of size $k+1$ whose deletion makes G_i bipartite $\Rightarrow$ We can use the compression algorithm to find a suitable S_i in time $O(3^k \cdot k|E(G_i)|)$.

Step 3: Iterative Compression

Bipartite-Deletion(G, k)

1. $S_k = \{v_1, \ldots, v_k\}$
2. for $i := k+1$ to n
3. Invariant: $G_{i-1} \setminus S_{i-1}$ is bipartite.
4. Call Compression($G_i, S_{i-1} \cup \{v_i\}$)
5. If the answer is "NO" $\Rightarrow$ return "NO"
6. If the answer is a set $X \Rightarrow S_i := X$
7. Return the set S_n

Running time: the compression algorithm is called n times and everything else can be done in linear time
$\Rightarrow O(3^k \cdot k|V(G)| \cdot |E(G)|)$ time algorithm.

Feedback Vertex Set in Tournaments

- **Input :** A tournament D and integer k.
- **Parameter :** k
- **Output :** $S \subseteq V(D)$ such that $D - S$ is an acyclic directed graph.

Feedback Vertex Set in Tournaments

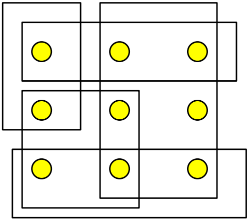
- **Input :** A tournament D and integer k.
- **Parameter :** k
- **Output :** $S \subseteq V(D)$ such that $D - S$ is an acyclic directed graph.

Tournament:

- A Complete Directed Graph.
- Exercise: There is a (directed) cycle if and only if there is a cycle of length 3.
- Exercise: Using iterative compression, show that FVST is FPT parameterized by the solution size k.

Dynamic Programming

Set Cover



- **Input :** A universe U of n elements and a family $\mathcal{F} = \{F_1, F_2, \ldots F_m\}$ of m subsets of U.
- **Parameter :** n (Size of the Universe)
- **Output :** A subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of minimum size that "covers U"

$$\bigcup_{F \in \mathcal{F}'} F = U$$

Theorem

SET COVER *is FPT parameterized by the size of the universe*

Running time: $2^n \cdot \mathsf{poly}(n, m)$

Set Cover: Dynamic Programming

- Fix an ordering of the family $\mathcal{F}$: $F_1, F_2, \ldots, F_m$
- Dynamic Programming Table,

$$\text{for every } X \subseteq U \text{ and } j \in \{0, 1, \ldots, m\}$$

$$T[X, j] = \text{ size of a min subset of } \{F_1, F_2, \ldots, F_j\} \text{ that covers } X$$

- Table Size: $2^{|U|} \times m$
- **Base Case :** $T[X, 0] = 0$ if $X = \emptyset$, else it is ∞.
- **Recursize Step :**

$$T[X, j] = \min \left\{ T[X, j-1],\ \ 1 + T[X \setminus F_j, j-1] \right\}$$

  - Either X can be covered using within $\{F_1, F_2, \ldots, F_{j-1}\}$
  - Or we need F_j + best solution of $X \setminus F_j$
- $T[U, m]$ is the minimum set cover size
- Maintain a candidate solution along with each $T[X, j]$.

Set Cover: Dynamic Programming

- Fix an ordering of the family $\mathcal{F}$: $F_1, F_2, \ldots, F_m$
- Dynamic Programming Table,

$$\text{for every } X \subseteq U \text{ and } j \in \{0, 1, \ldots, m\}$$

$$T[X, j] = \text{ size of a min subset of } \{F_1, F_2, \ldots, F_j\} \text{ that covers } X$$

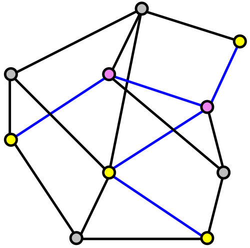
- Table Size: $2^{|U|} \times m$
- **Base Case :** $T[X, 0] = 0$ if $X = \emptyset$, else it is ∞.
- **Recursize Step :**

$$T[X, j] = \min \left\{ T[X, j-1], \;\; 1 + T[X \setminus F_j, j-1] \right\}$$

  - Either X can be covered using within $\{F_1, F_2, \ldots, F_{j-1}\}$
  - Or we need F_j + best solution of $X \setminus F_j$
- $T[U, m]$ is the minimum set cover size
- Maintain a candidate solution along with each $T[X, j]$.

Exercise: Prove that $T[X, j]$ indeed contains a minimum set cover of X from $\{F_1, F_2, \ldots, F_j\}$

Steiner Tree



- **Input :** Graph G on n vertices, $S \subseteq V(G)$ of k vertices called Terminals.
- **Parameter :** k (Number of Terminals)
- **Output :** Minimum connected subgraph H of G that contains all of S.

Observation : H must be a Tree

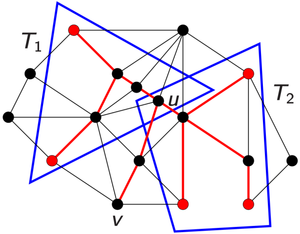
Theorem

Steiner Tree *can be solved in time* $3^k \cdot \mathsf{poly}(n)$.

Notation:

- $d_G(u, v) =$ length of shortest path between u and v in G.
- Assume every terminal $s \in S$ has degree 1

Steiner Tree: Dynamic Programming



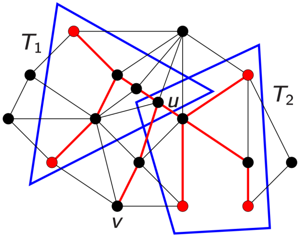
- DP Table: For $X \subseteq S$ and $v \in V(G)$

  $T[X, v] =$ minimum cost of a sub-tree containing $X \cup v$.

- Table Size: $2^S \cdot n$
- Base Case I: $T[\emptyset, v] = 1$ for every $v \in V(G)$
- Base Case II: $T[\{s\}, v] = d_G(s, v)$ for every $s \in S$
- Recursive Case: for $X \subseteq V(G)$, $|X| \geq 2$

$$T[X, v] = \min_{u \in V(G),\ \emptyset \neq Y \subsetneq X} d_G(u, v) + T[Y, u] + T[X \setminus Y, u]$$

Steiner Tree: Dynamic Programming



- DP Table: For $X \subseteq S$ and $v \in V(G)$

  $T[X, v] =$ minimum cost of a sub-tree containing $X \cup v$.

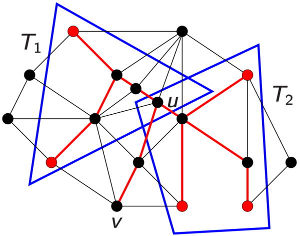
- Recursive Case:

$$T[X, v] = \min_{u \in V(G),\ \emptyset \neq Y \subsetneq X} d_G(u, v) + T[Y, u] + T[X \setminus Y, u]$$

Correctness: (LHS $\leq$ RHS)

- For any $Y \subseteq X$ and $u \in V(G)$, the RHS is the cost of a sub-tree connecting $X \cup v$.
- RHS = min-cost subtree for $Y \cup u$ + min-cost subtree for $(X \setminus Y) \cup u$ + shortest path between u and v

Steiner Tree: Dynamic Programming



- DP Table: For $X \subseteq S$ and $v \in V(G)$

  $T[X, v]$ = minimum cost of a sub-tree containing $X \cup v$.

- Recursive Case:

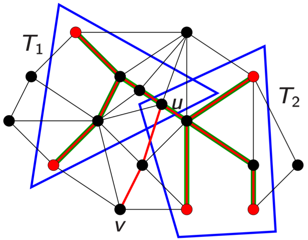
$$T[X, v] = \min_{u \in V(G),\ \emptyset \neq Y \subsetneq X} d_G(u, v) + T[Y, u] + T[X \setminus Y, u]$$

Correctness: (LHS $\geq$ RHS)

- Consider a minimum subtree H of G connecting $X \cup v$.
- root H at v, and u is the closest descendant with multiple children $\{u_1, u_2, \ldots, u_\ell\}$

Note: u exists because $|X| \geq 2$ and all terminals have degree 1.
Further $d_H(u, v) = d_G(u, v)$, by choice of H

Steiner Tree: Dynamic Programming



- DP Table: For $X \subseteq S$ and $v \in V(G)$

  $T[X, v] =$ minimum cost of a sub-tree containing $X \cup v$.

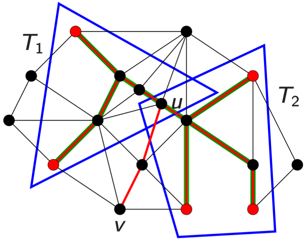
- Recursive Case:

$$T[X, v] = \min_{u \in V(G),\ \emptyset \neq Y \subsetneq X} d_G(u, v) + T[Y, u] + T[X \setminus Y, u]$$

Correctness: (LHS $\geq$ RHS)

- Let Y = all terminal from X in sub-tree of u_1.
- Split H into 3 parts
  - The sub-path between u and v
  - The sub-tree of H rooted at u_1 + edge (u, u_1)
  - The sub-tree of H excluding the above

Steiner Tree: Dynamic Programming



- DP Table: For $X \subseteq S$ and $v \in V(G)$

  $T[X, v] =$ minimum cost of a sub-tree containing $X \cup v$.

- Recursive Case:

$$T[X, v] = \min_{u \in V(G),\ \emptyset \neq Y \subsetneq X} d_G(u, v) + T[Y, u] + T[X \setminus Y, u]$$

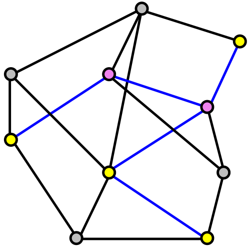
Running Time:

- Computing $T[X, v]$ requires $2^{|X|} \cdot \mathsf{poly}(n)$ time.
- Computing the entire table requires time:

$$\sum_{v \in V(G), X \subseteq S} 2^{|X|} \cdot \mathsf{poly}(n)$$

- This is $3^{|S|} \cdot \mathsf{poly}(n)$

Steiner Tree



- **Input :** Graph G on n vertices, $S \subseteq V(G)$ of k vertices called Terminals.
- **Parameter :** k (Number of Terminals)
- **Output :** Minimum connected subgraph H of G that contains all of S.

Observation : H must be a Tree

Theorem

Steiner Tree *can be solved in time* $3^k \cdot \mathsf{poly}(n)$.

Exercise: Steiner Tree with weights (Positive Integers)

Thank You.

Iterative Compression slides,
courtesy Neeldhara Misra and Daniel Marx.