FPT ALGORITHMS, PART III:
ITERATIVE COMPRESSION

Contents:
- A simple example: $k$-Vertex Cover
- Undirected Feedback Vertex Set
- Directed Feedback Vertex Set
Problem 1: Iterative compression for Vertex Cover

$k$-Vertex Cover ($k$-VC):
INSTANCE: A graph $G$ and integer $k$.
PARAMETER: $k$
QUESTION: Does $G$ have a vertex cover $S$ with $|S| \leq k$? (a $k$-VC.)
Lemma (Vertex Cover Compression)

If a \((k + 1)\)-VC \(S\) for a graph \(G\) on \(n\) vertices is given, then Algorithm 1 decides whether \(G\) has a \(k\)-VC in time \(2^k n^{O(1)}\).

(Of course, we already know multiple FPT algorithms that decide whether \(G\) has a \(k\)-VC, even without using \(S\), and even faster...)

Algorithm 1

INPUT: A \((k + 1)\)-VC \(S\) for graph \(G\).
OUTPUT: A \(k\)-VC for \(G\), or ‘NO’ if it does not exist.

For all \(S_{\text{keep}} \subseteq S\):

\[ S_{\text{rem}} := S \setminus S_{\text{keep}}. \]

Let \(S_{\text{new}}\) be all vertices in \(V \setminus S\) with neighbors in \(S_{\text{rem}}\).

If \(|S_{\text{new}}| + |S_{\text{keep}}| \leq k\) and \(G[S_{\text{rem}}]\) contains no edges, then

Return \(S_{\text{new}} \cup S_{\text{keep}}\)

Return ‘NO’
The correctness of Algorithm 1

**Proposition**

If $G[S_{\text{rem}}]$ contains no edges, then $S_{\text{new}} \cup S_{\text{keep}}$ is a VC for $G$.

**Proof:** Consider $uv \in E(G)$.

- If $u \in S_{\text{keep}}$ or $v \in S_{\text{keep}}$, the edge is covered.
- If $u \in V \setminus S$ and $v \in S_{\text{rem}}$ then $u \in S_{\text{new}}$.
- $u \in S_{\text{rem}}$ and $v \in S_{\text{rem}}$ is not possible by assumption.
- $u \in V \setminus S$ and $v \in V \setminus S$ is not possible since $S$ is a VC.

W.l.o.g. this covers all cases. □
Proposition

If $G$ contains a $k$-VC $S'$, then Algorithm 1 will return a $k$-VC.

Proof: Consider the iteration of the for-loop that considers $S_{\text{keep}} = S \cap S'$. Then $S_{\text{rem}} = S \setminus S'$.

Since $S'$ is a VC, $G[S_{\text{rem}}]$ contains no edges, and $S_{\text{new}} \subseteq S' \setminus S$.

Therefore, $|S_{\text{new}}| + |S_{\text{keep}}| \leq |S \cap S'| + |S' \setminus S| = |S'| \leq k$, so Algorithm 1 returns a solution in this iteration. \qed
The complexity of Algorithm 1

Proposition

 Algorithm 1 terminates in $2^{k+1} n^{O(1)}$ steps.

Proof: The for-loop considers $2^{k+1}$ different subsets of $S$. The steps in each iteration can easily be done in polynomial time. □

Now we have proved the compression lemma:

Lemma (Vertex Cover Compression)

If a $(k + 1)$-VC $S$ for a graph $G$ on $n$ vertices is given, then Algorithm 1 decides whether $G$ has a $k$-VC in time $2^k n^{O(1)}$. 
General Iterative Compression - Vertex Cover:

Let $V(G) = \{v_1, \ldots, v_n\}$, $G_i = G[v_1, \ldots, v_i]$ for all $i \in \{1, \ldots, n\}$.

$S_1 = \emptyset$ (*$S_1$ is a VC for $G_1$.*)

For $i := 1 \ldots n - 1$:

If $S_i$ is a VC for $G_{i+1}$ then $S_{i+1} := S_i$ else $S_{i+1} := S_i \cup \{v_{i+1}\}$

If $|S_{i+1}| > k$:

Use an FPT algorithm $\mathbb{A}$ to compute a $k$-VC $S_{i+1}$ for $G_{i+1}$,

or return ‘NO’ and halt if it does not exist.

Return ‘YES’.

- If Algorithm $\mathbb{A}$ has complexity $f(k) \cdot g(n)$, then the above algorithm correctly decides $k$-VC in time $f(k) \cdot g(n) \cdot O(n)$.
Let $V(G) = \{v_1, \ldots, v_n\}$, $G_i = G[v_1, \ldots, v_i]$ for all $i \in \{0, \ldots, n\}$.

$S_1 = \emptyset$ ($S_1$ is a VC for $G_1$.)

For $i := 1 \ldots n - 1$:

- If $S_i$ is a VC for $G_{i+1}$ then $S_{i+1} := S_i$ else $S_{i+1} := S_i \cup \{v_{i+1}\}$

- If $|S_{i+1}| > k$:
  - For all $S_{\text{keep}} \subset S_{i+1}$:
    - Decide whether a $k$-VC $S'$ with $S' \cap S_{i+1} = S_{\text{keep}}$ exists.
    - If $S'$ is such a $k$-VC then
      $S_{i+1} := S'$, break (out of the top level for loop).
  - If $|S_{i+1}| > k$ then return ‘NO’, halt.

Return ‘YES’.
General Iterative Compression:

(Q) For what kind of problems is this approach suitable?

(A) For any *minimization* problem on instances $G$, with integer objective value, where we can construct a sequence $G_1, \ldots, G_n$ of *polynomial length* with $G_n = G$ and:

(1) A $k$-solution for $G_1$ exists and can be found in polynomial time. (A *$k$-solution* is a feasible solution with objective value at most $k$.)

(2) If $G_i$ has a $k$-solution, then $G_{i+1}$ has a $(k+1)$-solution, which can be found in polynomial time.

(3) If $G_{i+1}$ has a $k$-solution, then $G_i$ has a $k$-solution.

(4) If a $(k+1)$-solution $S$ for $G_{i+1}$ is given, then there is an FPT algorithm for parameter $k$ that decides whether $G_{i+1}$ has a $k$-solution (*The compression step*).
For problems that satisfy the properties of the previous slide, the following algorithm is an FPT algorithm for parameter $k$ that decides whether a $k$-solution exists for $G$:

Let $S_1$ be $k$-solution for $G_1$.
For $i := 1 \ldots n - 1$:
   Using $S_i$, construct a $(k + 1)$-solution $S_{i+1}$ for $G_{i+1}$.
   Use an FPT algorithm $A$ to compute a $k$-solution $S_{i+1}$ for $G_{i+1}$, or return ‘NO’ and halt if it does not exist.
Return ‘YES’
Problem 2: (Undirected) Feedback Vertex Set

- A **cycle** in an (undirected) graph $G$ is a connected subgraph $C$ in which all vertices have degree 2.

- A **feedback vertex set** for $G$ is a set $S \subseteq V(G)$ such that $G - S$ contains no cycles.

$k$-Feedback Vertex Set (**$k$-FVS**):

INSTANCE: A graph $G$ and integer $k$.
PARAMETER: $k$
QUESTION: Does there exist a FVS $S$ for $G$ with $|S| \leq k$? (a $k$-FVS.)
Let $V(G) = \{v_1, \ldots, v_n\}$, and define again $G_i = G[v_1, \ldots, v_i]$ for all $i \in \{1, \ldots, n\}$.

$k$-FVS satisfies the easy conditions for applying iterative compression:

1. $S_1 = \emptyset$ is a $k$-FVS for $G_1$.

2. If $S_i$ is a $k$-FVS for $G_i$, then $S_i \cup \{v_{i+1}\}$ is a $(k + 1)$-FVS for $G_{i+1}$.

3. If $S_{i+1}$ is a $k$-FVS for $G_{i+1}$, then $S_{i+1} \setminus \{v_{i+1}\}$ is a $k$-FVS for $G_i$.

(Q) But can we prove a compression lemma for it?
We want to prove:

**Lemma (FVS Compression)**

*If a \((k + 1)\)-FVS \(S\) for \(G\) is given, then there is an FPT algorithm for parameter \(k\) that decides whether \(G\) has a \(k\)-FVS.*

- Again, assuming a \(k\)-FVS \(S'\) exists, we will try all possibilities for \(S_{\text{keep}} = S' \cap S\), and in one case we ‘guess correctly’.

**Observation**

*\(G\) has a \(k\)-FVS \(S'\) with \(S' \cap S = S_{\text{keep}}\) if and only if \(G - S_{\text{keep}}\) has a \(k'\)-FVS \(S''\) with \(S'' \cap S = \emptyset\), where \(k' = k - |S_{\text{keep}}|\).*
Proof of the observation

Proof: \(\Rightarrow\): Let \(S'\) be a \(k\)-FVS for \(G\) with \(S' \cap S = S_{\text{keep}}\).
Consider \(S'' = S' \setminus S_{\text{keep}}\).

- \(G - S_{\text{keep}} - S'' = G - S'\) contains no cycles, so \(S''\) is a FVS for \(G - S_{\text{keep}}\).
- \(|S''| = |S'| - |S_{\text{keep}}| \leq k - |S_{\text{keep}}|\) so it is a \(k'\)-FVS.
- \(S'' = S' \setminus S_{\text{keep}} = S' \setminus (S' \cap S) = S' \setminus S\) so \(S \cap S'' = \emptyset\).
This proves that \(S''\) is a \(k'\)-FVS for \(G - S_{\text{keep}}\) with \(S'' \cap S = \emptyset\).

\(\Leftarrow\): Let \(S''\) be a \(k'\)-FVS for \(G - S_{\text{keep}}\) with \(S'' \cap S = \emptyset\).
Consider \(S' = S'' \cup S_{\text{keep}}\).

- \(G - S' = G - S_{\text{keep}} - S''\) contains no cycles, so \(S'\) is a FVS for \(G\).
- \(|S'| = |S''| + |S_{\text{keep}}| \leq k' + |S_{\text{keep}}| = k\).
- \(S' \cap S = S_{\text{keep}}\), since \(S'' \cap S = \emptyset\).
This proves that \(S'\) is a \(k\)-FVS for \(G\) with \(S' \cap S = S_{\text{keep}}\). \(\square\)
Observation

Let $S$ be a FVS for $G$, and $S_{\text{keep}} \subseteq S$. Then $S_{\text{rem}} = S \setminus S_{\text{keep}}$ is a FVS for $G' = G - S_{\text{keep}}$.

Therefore we have reduced the problem to: given a FVS $S_{\text{rem}}$ for $G'$ with $|S_{\text{rem}}| = k' + 1$, is there a $k'$-FVS $S_{\text{new}}$ for $G'$ with $S_{\text{new}} \cap S_{\text{rem}} = \emptyset$?
**k-S-Disjoint FVS:**

**INSTANCE:** A graph \( G \), FVS \( S \) for \( G \), and integer \( k = |S| - 1 \).

**PARAMETER:** \( k \)

**QUESTION:** Is there a \( k \)-FVS \( S' \) for \( G \) with \( S' \cap S = \emptyset \)? (A \( k \)-S-disjoint FVS.)

**Goal:** give a *kernelization* for this problem.
Observation (Degree 1)

Let $v \in V(G) \setminus S$ have degree 1. Then $G$ has a $k$-$S$-disjoint FVS iff $G - v$ has a $k$-$S$-disjoint FVS.
Reduction Rules - Degree 2

Now let $u \in V(G) \setminus S$ with $N(u) = \{v, w\}$.

Proposition (Degree 2 A)
If $vw \in E(G)$ and $w \notin S$, then $G$ has a $k$-$S$-disjoint FVS iff $G - w$ has a $(k - 1)$-$S$-disjoint FVS.

Proposition (Degree 2 B)
If $vw \notin E(G)$ and $w \notin S$, then $G$ has a $k$-$S$-disjoint FVS iff $G'$ has a $k$-$S$-disjoint FVS, where $G'$ is obtained by contracting $uw$.

Conclusion: in a reduced instance $G, S, k$, all vertices of $V(G) \setminus S$ have degree at least 3, or only neighbors in $S$. 
Definitions

**Goal:** Let $G, S, k$ be a reduced $k$-$S$-disjoint FVS instance, and suppose a $k$-$S$-disjoint FVS $S'$ exists. Prove an upper bound on $|V(G)|$.

- Let $T = G - S$.
- $T$ is a *forest*, which means that every component is a tree.
- Let $H$ be the vertices $v \in V(T)$ with $d_T(v) \geq 3$. (*High degree.*)
- Let $L$ be the vertices $v \in V(T)$ with $d_T(v) = 1$. (*Leaves.*)
- Let $R$ be the vertices $v \in V(T)$ with $d_T(v) = 2$. (*Rest.*)
- Let $Z$ be the vertices $v \in V(T)$ with $d_T(v) = 0$. (*Zero.*)
- Let $H' = H \cap S'$, $L' = L \cap S'$, $R' = R \cap S'$ and $Z' = Z \cap S'$. 
A lower bound for $|E(T - S')|$ 

- By $|E(T)| = \frac{1}{2} \sum_{v \in V(T)} d_T(v)$, we have

$$|E(T)| = \frac{1}{2} |L| + |R| + \frac{1}{2} \sum_{v \in H} d_T(v).$$

- The number of edges removed by deleting $S' = H' \cup L' \cup R'$ is at most

$$|L'| + 2|R'| + \sum_{v \in H'} d_T(v).$$

Proposition

Let $T$ be a forest with $c$ (tree) components, where the set $H$ ($L$) contains the vertices with degree at least 3 (exactly 1), respectively. Then $\sum_{v \in H} (d(v) - 2) = |L| - 2c$. 
Combining the previous (in)equalities gives:

\[ |E(T - S')| \geq \]

\[ \frac{1}{2}|L| + |R| + \frac{1}{2} \sum_{v \in H} d_T(v) - |L'| - 2|R'| - \sum_{v \in H'} d_T(v) \geq \]

\[ \sum_{v \in H} (d_T(v) - 1) + |R| - |L'| - 2|R'| - \sum_{v \in H'} d_T(v) = \]

\[ \sum_{v \in H \setminus H'} d_T(v) - |H| + |R| - |L'| - 2|R'| \geq \]

\[ 3|H \setminus H'| - |H| + |R| - |L'| - 2|R'| = \]

\[ 2|H| - 3|H'| + |R| - 2|R'| - |L'|. \]
Lower and upper bounds for $|E(G - S')|$

- Because $G$ is reduced, vertices in $L$, $R$ and $Z$ have at least two, one and two neighbors in $S$, respectively. So:

$$|E(G - S')| \geq |E(T - S')| + 2|L \setminus L'| + |R \setminus R'| + 2|Z \setminus Z'| \geq 2|H| - 3|H'| + |R| - 2|R'| - |L'| + 2|L| - 2|L'| + |R| - |R'| + 2|Z| - 2|Z'| = 2|H| - 3|H'| + 2|R| - 3|R'| + 2|L| - 3|L'| + 2|Z| - 2|Z'|.$$

- Since $S'$ is a FVS, $F = G - S'$ is a forest, so

$$|E(G - S')| \leq |V(F)| - 1 = |S| + |H| + |L| + |R| + |Z| - |S'| - 1.$$
Combining these bounds gives:

\[ 2|H| - 3|H'| + 2|R| - 3|R'| + 2|L| - 3|L'| + 2|Z| - 2|Z'| \leq \]

\[ |E(G - S')| \leq \]

\[ |S| + |H| + |L| + |R| + |Z| - |S'| - 1 \iff \]

\[ |H| + |L| + |R| + |Z| \leq 2|S'| + |S| - 1 \iff |V(G) \setminus S| \leq 3k. \]

(The whole graph (kernel) has at most 4k + 1 vertices.)
An FPT Algorithm for $k$-$S$-disjoint FVS

**Theorem**

For a graph $G$ on $n$ vertices that has a FVS $S$ with $|S| = k + 1$, it can be decided whether $G$ has a $k$-$S$-disjoint FVS in time $n^{O(1)} + O^*(6.75^k)$.

**Algorithm:**

1. If $G[S]$ contains a cycle then return ‘NO’.
2. In time $n^{O(1)}$, apply the degree 1 and 2 reduction rules until a reduced instance $G'$, $S$, $k'$ is obtained.
3. If $|V(G') \setminus S| > 3k$ then return ‘NO’.
4. Otherwise, test all subsets $S' \subseteq V(G') \setminus S$ with $|S'| = k'$. If one of these is a FVS then return ‘YES’, else return ‘NO’.

- The correctness of Step (3) follows from the bound we proved for reduced instances. The correctness of the other steps is obvious.

**Q** How to prove the complexity bound?
Intermezzo: Stirling’s Appr. for Computer Scientists

Theorem (Stirling’s approximation)
\[ \lim_{n \to \infty} \frac{n!}{\sqrt{2\pi n (\frac{n}{e})^n}} = 1. \]

Corollary
\[ n! \in \Theta \left( \sqrt{n} \cdot \left( \frac{n}{e} \right)^n \right). \]

Recall:
- \( f(n) \in \Omega(g(n)) \) means: there exist \( c \) and \( N \) such that for all \( n > N \), \( |f(n)| \geq cg(n) \).
- \( f(n) \in \Theta(g(n)) \) means: \( f(n) \in O(g(n)) \) and \( f(n) \in \Omega(g(n)) \).
- \( f(n) \in \Theta(g(n)) \) is also written as \( f(n) \sim g(n) \).
- Observe: if \( f_1(n) \sim g_1(n) \) and \( f_2(n) \sim g_2(n) \), then \( f_1(n) \cdot f_2(n) \sim g_1(n) \cdot g_2(n) \) and \( f_1(n)/f_2(n) \sim g_1(n)/g_2(n) \).
Complexity of the algorithm

Applying the reduction rules takes time $n^{O(1)}$.

Let the reduced instance $G'$ have $n'$ vertices not in $S$. If $n' \leq 3k'$ \textit{(and $k \geq 2$)}, then the number of sets tested is:

$$\binom{n'}{k'} \leq \binom{3k'}{k'} \leq \binom{3k}{k} = \frac{(3k)!}{(2k)!k!} \sim$$

$$\frac{\sqrt{3k(3k)^3 e^{-3k}}}{\sqrt{2k(2k)^2 e^{-2k}}} \cdot \sqrt{kk^k e^{-k}} <$$

$$\frac{3^{3k}}{2^{2k}} = \left(\frac{3^3}{2^2}\right)^k = 6.75^k.$$

Testing whether a set $S'$ is a FVS can be done in polynomial time $k^{O(1)}$, so the total complexity is $n^{O(1)} + O(6.75^k) \cdot k^{O(1)}$. 
Intermezzo: $O^*$-notation

On the last slide we had a function $f(k) \in \Theta(6.75^k \cdot k^c)$ for some constant $c$.

- Observe that: $f(k) \in O(6.7500001^k)$, but $f(k) \notin O(6.75^k)$.

- So, the polynomial factor $k^c$ in the bound on $f(k)$ seems irrelevant, but still we may not just omit it using the $O$-notation. To remedy this annoying situation, the $O^*$-is defined:

  - We may write $f(n) \in O^*(g(n))$ if:
    - $g(n) \in 2^{\Omega(n)}$, and
    - there exists a $c$ such that $f(n) \in O(g(n) \cdot n^c)$.

- The previous algorithm has complexity $n^{O(1)} + O^*(6.75^k)$.
- This notation is useful but often used incorrectly, so be careful!
We need to solve the following compression problem at most \( n \) times (once for every \( G_i \)):
given a \((k + 1)\)-FVS \( S \) for \( G_i \), can we find a \( k \)-FVS \( S' \) for \( G_i \)?

This is done by considering \( 2^{k+1} \) ‘guesses’ for \( S_{\text{keep}} = S \cap S' \). For each such guess, we need to solve a \( k'\)-\( S_{\text{rem}} \)-disjoint FVS problem, with \( k' \leq k \).

This last problem can be solved in time \( n^{O(1)} + O^*(6.75^k) \).

**Theorem**

\( k \)-FVS can be decided in time \( n^{O(1)}O^*(13.5^k) \).
Problem 3: Directed Feedback Vertex Set

- A cycle in an digraph $G$ is a connected subgraph $C$ in which all vertices have in-degree 1 and out-degree 1.

- A feedback vertex set for a digraph $G$ is a set $S \subseteq V(G)$ such that $G - S$ contains no cycles.

$k$-Directed Feedback Vertex Set ($k$-DFVS):
INSTANCE: A digraph $G$ and integer $k$.
PARAMETER: $k$
QUESTION: Does there exist a FVS $S$ for $G$ with $|S| \leq k$? (a $k$-DFVS.)
• We use standard iterative compression again. Similar to before, this reduces \(k\)-DFVS to the following problem:

**\(k\)-S-Disjoint DFVS:**

INSTANCE: A digraph \(G\), FVS \(S\) for \(G\) and integer \(k = |S| - 1\).

PARAMETER: \(k\).

QUESTION: Is there a \(k\)-FVS \(S'\) for \(G\) with \(S' \cap S = \emptyset\)?

(A **\(k\)-S-disjoint DFVS.**)

• At this point, a new approach is needed: no kernelization algorithm is known for \(k\)-S-Disjoint DFVS.
Acyclic digraphs

- A digraph without cycles is called **acyclic**.
  
  *(Note that when replacing (directed) arcs with (undirected) edges, the resulting graph may have many (undirected) cycles.)*

**Proposition**

An acyclic digraph contains a vertex with in-degree 0, and a vertex with out-degree 0.

**Proposition**

*The vertices of an acyclic digraph \( G \) can be numbered \( v_1, \ldots, v_n \) such that for all \( i < j \), \( v_i \) is not reachable from \( v_j \) in \( G \).*

- Such a vertex order is called a **topological ordering** of \( V(G) \).
One final transformation

Let $G, S, k$ be a $k$-$S$-Disjoint DFVS instance, with $G = (V, E)$.

A bijective function $\sigma: \{1, \ldots, k + 1\} \to S$ is called a **numbering** of $S$.

- For any numbering $\sigma$ of $S$, the graph $G'(\sigma)$ is obtained by starting with $G[V \setminus S]$, and for every $i \in \{1, \ldots, k + 1\}$:
  - adding vertices $s_i$ and $t_i$, and
  - for every out-neighbor $v$ of $\sigma(i)$ adding an arc $(s_i, v)$, and
  - for every in-neighbor $v$ of $\sigma(i)$ adding an arc $(v, t_i)$ (if it was not already added).

**Lemma**

$S' \subseteq V \setminus S$ is a $k$-$S$-disjoint DFVS for $G$ if and only if there exists a numbering $\sigma$ of $S$ such that $S'$ is a $k$-$SkSep$ for $G'(\sigma)$. 
Proof direction 1: a DFVS is a SkSep for some $G'(\sigma)$.

Proof: Let $S'$ be a $k$-S-Disjoint DFVS for $G$. $G - S'$ is acyclic, so it admits a topological ordering $v_1, \ldots, v_m$.

All vertices of $S$ are part of $G - S'$, so this order can be used to define a numbering $\sigma$ of $S$ such that for all $j < i$, $\sigma(j)$ is not reachable from $\sigma(i)$.

Consider $G' = G'(\sigma)$. Suppose $G' - S'$ contains an $(s_i, t_j)$-path for $j \leq i$.

If $j < i$, this gives a $(\sigma(i), \sigma(j))$-path in $G - S'$, contradicting the choice of $\sigma$.

If $j = i$, this gives a cycle in $G - S'$ (containing $\sigma(i)$), contradiction.

Therefore, $S'$ is a SkSep for $G'$. 
Proof direction 2: a SkSep for some $G'(\sigma)$ is a DFVS.

Let $S'$ be a $k$-SkSep for $G' = G'(\sigma)$.

Suppose $G - S'$ contains a cycle $C$. Since $S$ is a FVS for $G$, $C$ contains at least one vertex of $S$.

If $C$ contains exactly one vertex $\sigma(i)$ of $S$, then $C$ corresponds to an $(s_i, t_i)$-path in $G' - S'$, contradiction.

If $C$ contains $l \geq 2$ vertices of $S$, then let $\sigma(i_0), \sigma(i_1), \ldots, \sigma(i_{l-1})$ be the order of those vertices in $C$.

Note that $i_x$ and $i_y$ are distinct when $x \neq y$. Therefore, there exists an $x$ such that $i_x > i_{(x+1) \mod l}$. Let $i = i_x$ and $j = i_{(x+1) \mod l}$.

The subpath of $C$ from $\sigma(i)$ to $\sigma(j)$ corresponds to an $(s_i, t_j)$-path in $G' - S'$ with $j < i$, contradiction. \qed
An FPT Algorithm for Directed Feedback Vertex Set: Summary

- We need to solve the following compression problem at most $n$ times (once for every $G_i$): given a $(k + 1)$-DFVS $S$ for $G_i$, can we find a $k$-DFVS $S'$ for $G_i$?

- This is done by considering all possible ‘guesses’ for $S_{\text{keep}} = S \cap S'$. For each such guess, we need to solve an $l$-$S^*$-Disjoint DFVS problem, with $l = k - |S_{\text{keep}}| \leq k$.

- For each $l$-$S^*$-Disjoint DFVS problem that needs to be solved, we consider $(l + 1)!$ ‘guesses’ for the numbering $\sigma$ of $S^*$. For each such guess, we need to solve a $l$-SkSep Problem.

- This last problem can be solved in time $n^{O(1)} \cdot 4^l$. 
Parameter function of the combined algorithm:

\[
\sum_{l=0}^{k} \binom{k+1}{l+1}(l+1)!4^l = 
\]

\[
\sum_{l=0}^{k} \frac{(k+1)!}{(k-l)!}4^l < (k+1)!(k+1)4^k.
\]

**Theorem**

\(k\)-DFVS can be decided in time \(n^{O(1)} \cdot O^*(4^k \cdot (k+1)!).\)