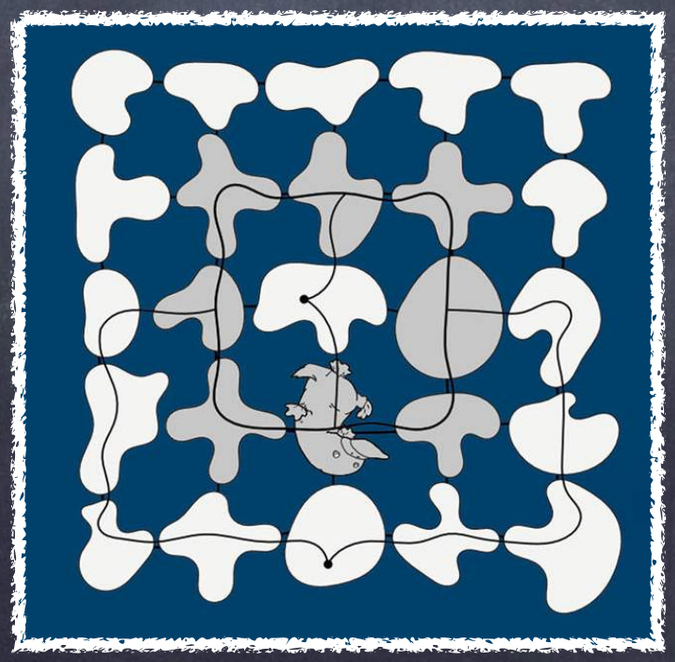
Basics II

Fedor V. Fomin
University of Bergen



Recent Advances in Parameterized Complexity

DECEMBER 3-7, 2017, Tel Aviv, Israel

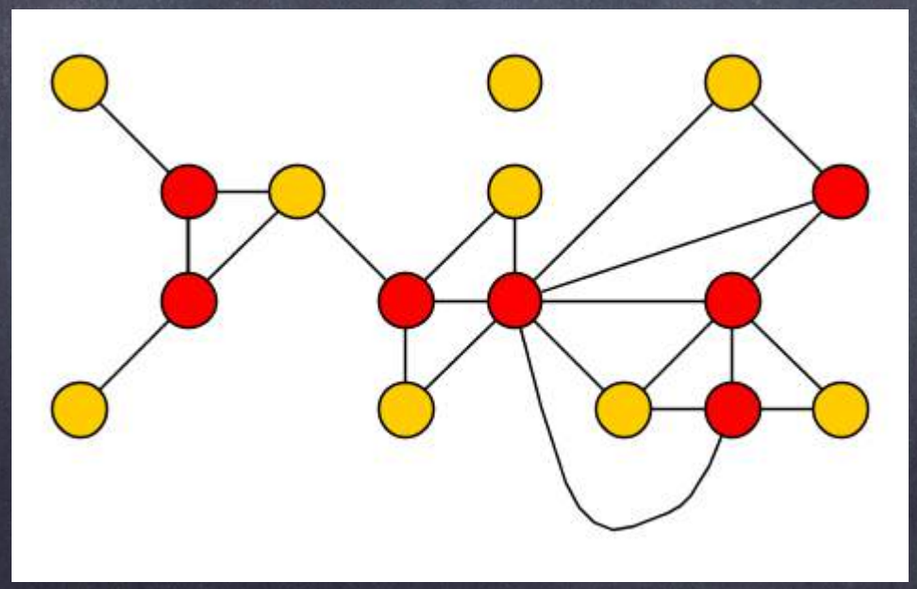
Plan

- Bounded Search Trees aka Branching
  - Vertex Cover: 1.62^k
  - 3-Hitting Set: 3^k
  - Feedback Vertex Set in Tournaments (FVST): 3^k
- Iterative Compression
  - FVST: 2^k
  - 3-Hitting Set: $(1+VC)^k$

Bounded Search Trees

Vertex Cover

Input: A graph G and an integer k
Parameter: k
Question: Is there a set S of at most k vertices in G, such that each edge has an endpoint in S?



Vertex Cover

Observation. Let uv be an edge of a graph G. Then for every vc S of G, either u or v is in S.

In other words, (G,k) is a yes-instance of Vertex Cover if and only if either $(G-u, k-1)$ or $(G-v, k-1)$ is a yes-instance.

Vertex Cover

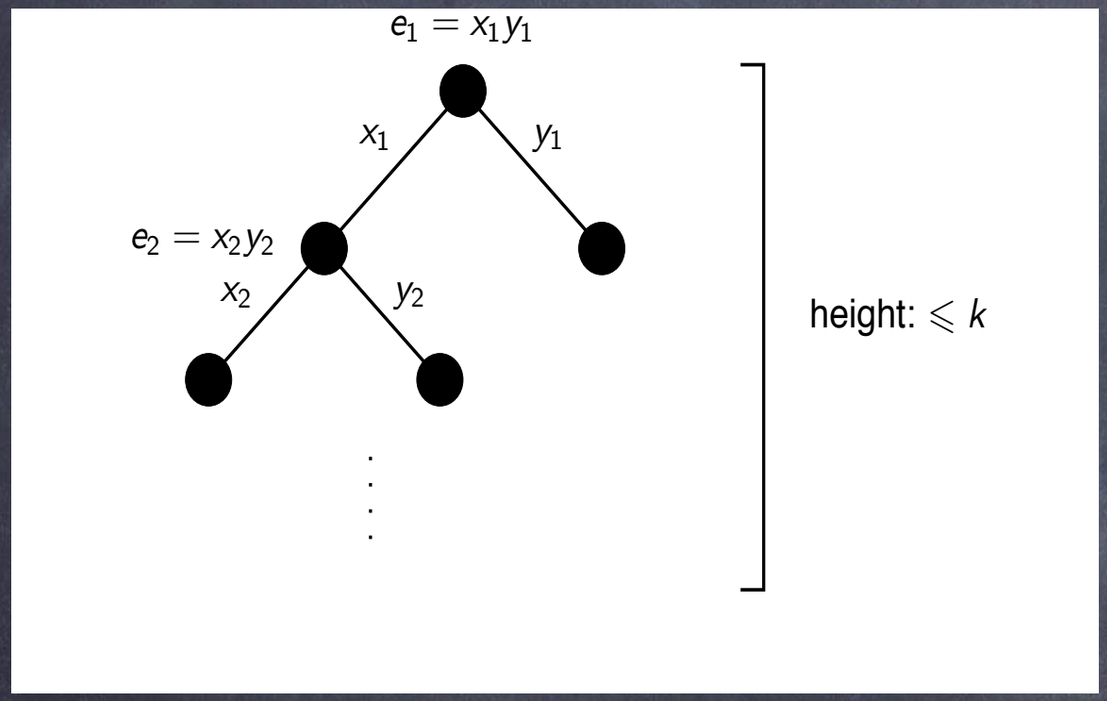
Algorithm VC(G,k)

- If k<0, output NO
- If G has no edges, output YES
- Let uv be an edge,

  return VC(G-u, k-1) ∨ VC(G-v, k-1)

Vertex Cover

Bounded search (branching) tree

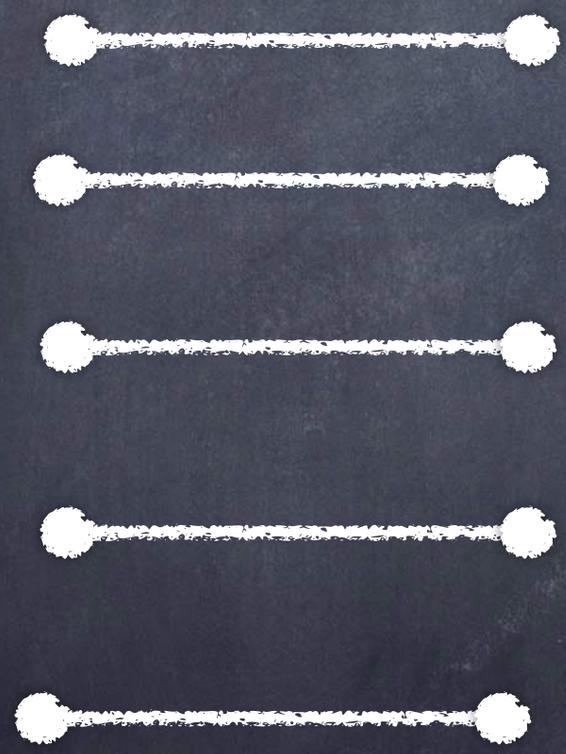


Running time proportional to the number of vertices in the branching tree

- $T(k) = 2^k \, \text{poly}(n)$

Vertex Cover

Worst-case example: k matchings



On the other hand, when all vertices are of degree at most 1, Vertex Cover is solvable in polynomial time

Vertex Cover

Observation. Let v be a vertex of degree at least 1 of G. Then for every vc S of G, either v or $N(v)$ is in S.

In other words, (G,k) is a yes-instance of Vertex Cover if and only if either $(G-v, k-1)$ or $(G-N(v), k-|N(v)|)$ is a yes-instance.

Vertex Cover

Algorithm VC2(G,k)

- If k<0, output NO
- If G has no vertex of degree at least 2, solve the problem in polynomial time
- Let v be a vertex of degree at least 2

  return VC2(G-v, k-1) ∨ VC2(G-N(v), k-|N(v)|)

Vertex Cover

Running time

$$T(k) = \begin{cases} T(k-1) + T(k-2) & \text{if } k \geq 2, \\ 1 & \text{otherwise,} \end{cases}$$

$T(k) \leq c \cdot \lambda^k$ $\qquad$ $c \cdot \lambda^k \geq c \cdot \lambda^{k-1} + c \cdot \lambda^{k-2}$

$$T(k) = T(k-1) + T(k-2) \leq c \cdot \lambda^{k-1} + c \cdot \lambda^{k-2} \leq c \cdot \lambda^k$$

Vertex Cover

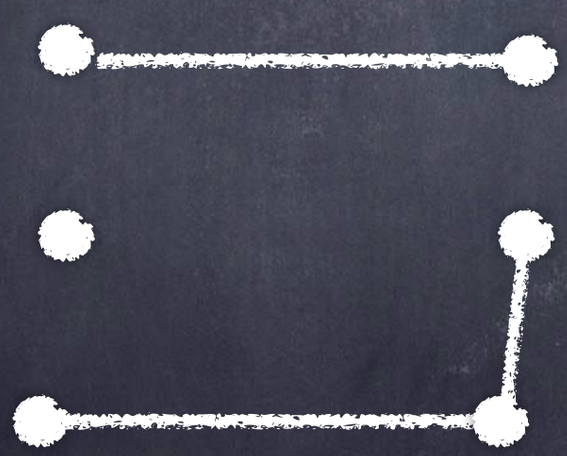
Running time

$$T(k) = T(k-1) + T(k-2) \leq c \cdot \lambda^{k-1} + c \cdot \lambda^{k-2} \leq c \cdot \lambda^{k}$$

$$\lambda^2 \geq \lambda + 1$$

$$\lambda = \frac{1+\sqrt{5}}{2} < 1.6181 \quad \longrightarrow \quad T(k) \leq 1.6181^k$$

Vertex Cover



On the other hand, when all vertices are of degree at most 2, Vertex Cover is solvable in polynomial time

Vertex Cover

Running time

$$T(k) = \begin{cases} T(k-1) + T(k-3) & \text{if } k \geq 3, \\ 1 & \text{otherwise.} \end{cases}$$

$$\lambda^3 = \lambda^2 + 1$$

Vertex Cover is solvable in time $1.4656^k n^{O(1)}$

Vertex Cover

Vertex Cover is solvable in time $1.4656^k n^{O(1)}$

Is this bound tight?

There are two questions:

Can the function T(k) be that large?

- Yes (ignoring rounding problems).

Can the bounded search tree of the Vertex Cover algorithm be that large?

- Difficult question, hard to answer in general.

Remark: the current record for VC 1.2738^k

More generally

Linear recurrence

$$T(k) = T(k - d_1) + T(k - d_2) + \cdots + T(k - d_p)$$

$$\lambda^k \geq \lambda^{k-d_1} + \lambda^{k-d_2} + \cdots + \lambda^{k-d_p}$$

Characteristic polynomial

$$\lambda^d - \lambda^{d-d_1} - \lambda^{d-d_2} - \ldots - \lambda^{d-d_p} \geq 0$$

Unique root: branching number corresponding to branching vector

$$(d_1, d_2, \ldots, d_p)$$

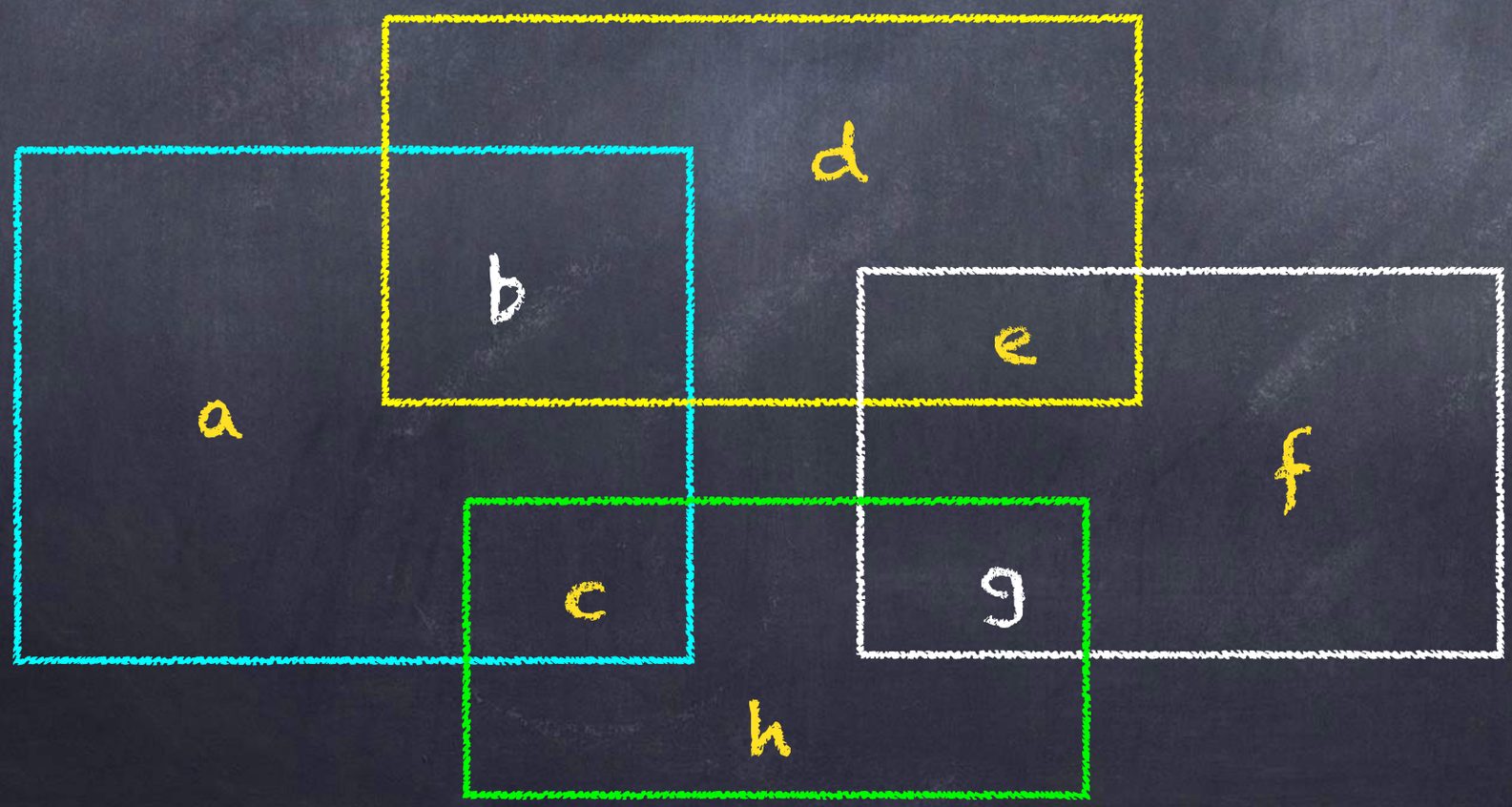
(i,j)	1	2	3	4	5	6
1	2.0000	1.6181	1.4656	1.3803	1.3248	1.2852
2		1.4143	1.3248	1.2721	1.2366	1.2107
3			1.2560	1.2208	1.1939	1.1740
4				1.1893	1.1674	1.1510
5					1.1487	1.1348
6						1.1225

Table 1: A table of branching numbers (rounded up)

d-Hitting Set

Input: A universe U, a family A of sets of size d over U, integer k

Question: Does there exist a subset X of U of size k that has a nonempty intersection with every member of A?



3-Hitting Set

Algorithm: Similar to Vertex Cover=2-Hitting Set

- Let uv be an edge, return $VC(G-u, k-1) \vee VC(G-v, k-1)$

Which is the same as

- Let uv be an edge, return $VC(G-E(u), k-1) \vee VC(G-E(v), k-1)$

3-Hitting Set

Algorithm: 3-HS(A, k)

- If A is empty, say YES
- If k<0, say NO
- Let uvw be a set from A, return
- 3-HS(A-S(u), k-1) ∨ 3-HS(A-S(v), k-1) ∨ 3-HS(A-S(w), k-1)

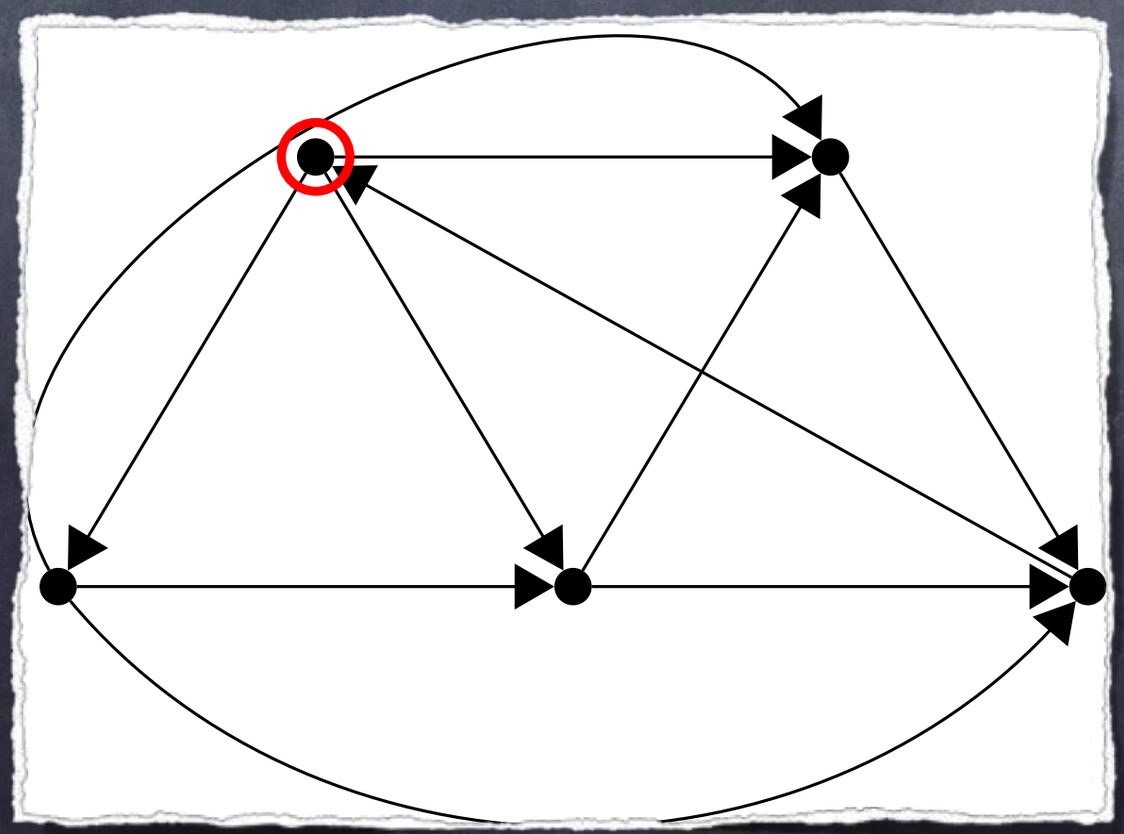
Running time

- 3^k poly(n)

Feedback Vertex Set in Tournament (FVST)

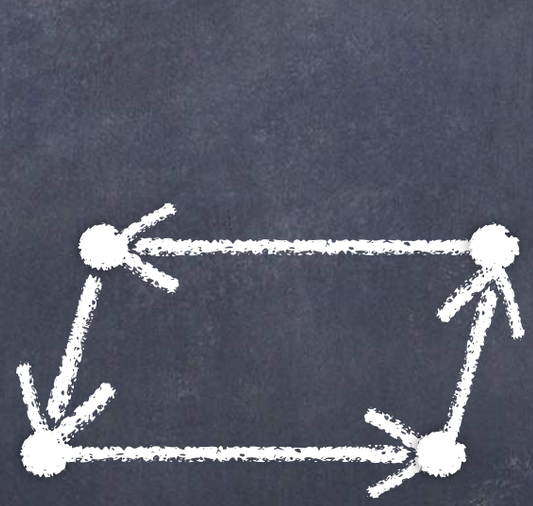
Input: A tournament (oriented clique) T and an integer k

Question: Does there exist a subset X of $V(T)$ of size k such that $T-X$ is acyclic?



FVST

Observation: A tournament is acyclic if and only if it does not contain a directed triangle.



FVST

Hitting directed triangles

3-Hitting Set

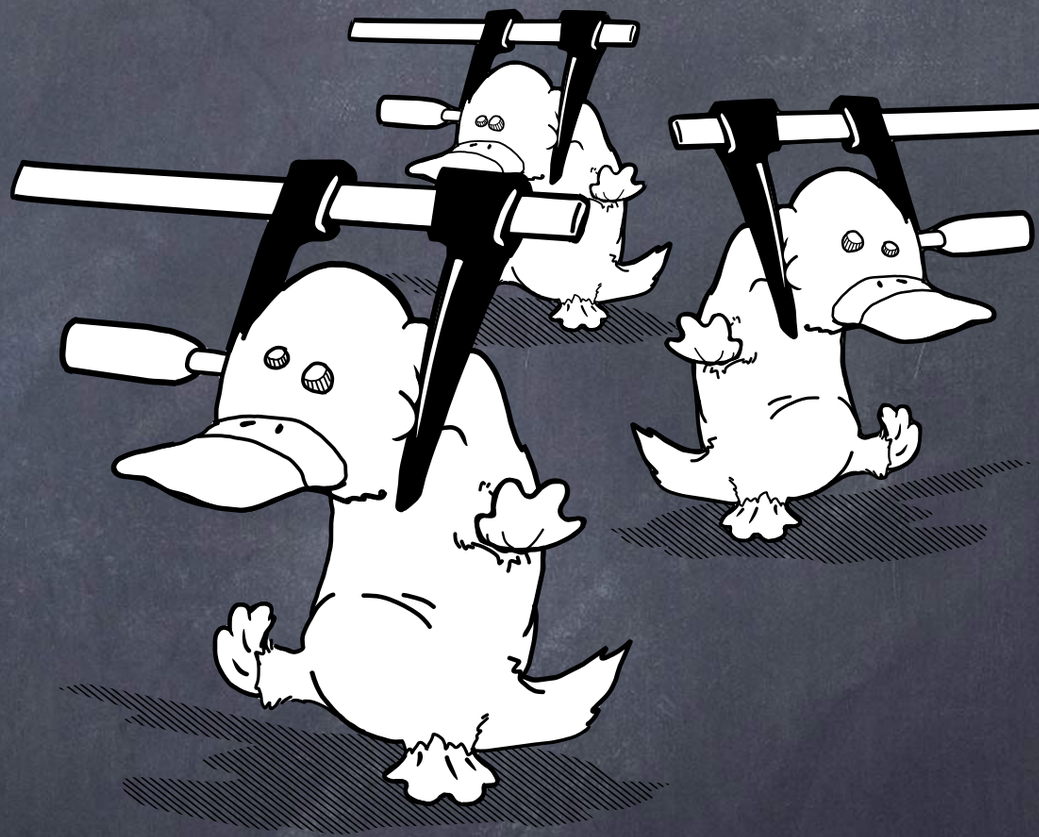
Running time 3^k poly(n)

More problems to branch...

Exercises

- Cluster Vertex Deletion: 3^k poly(n)
- Minimum Fill-in: k^k poly(n)

Iterative Compression



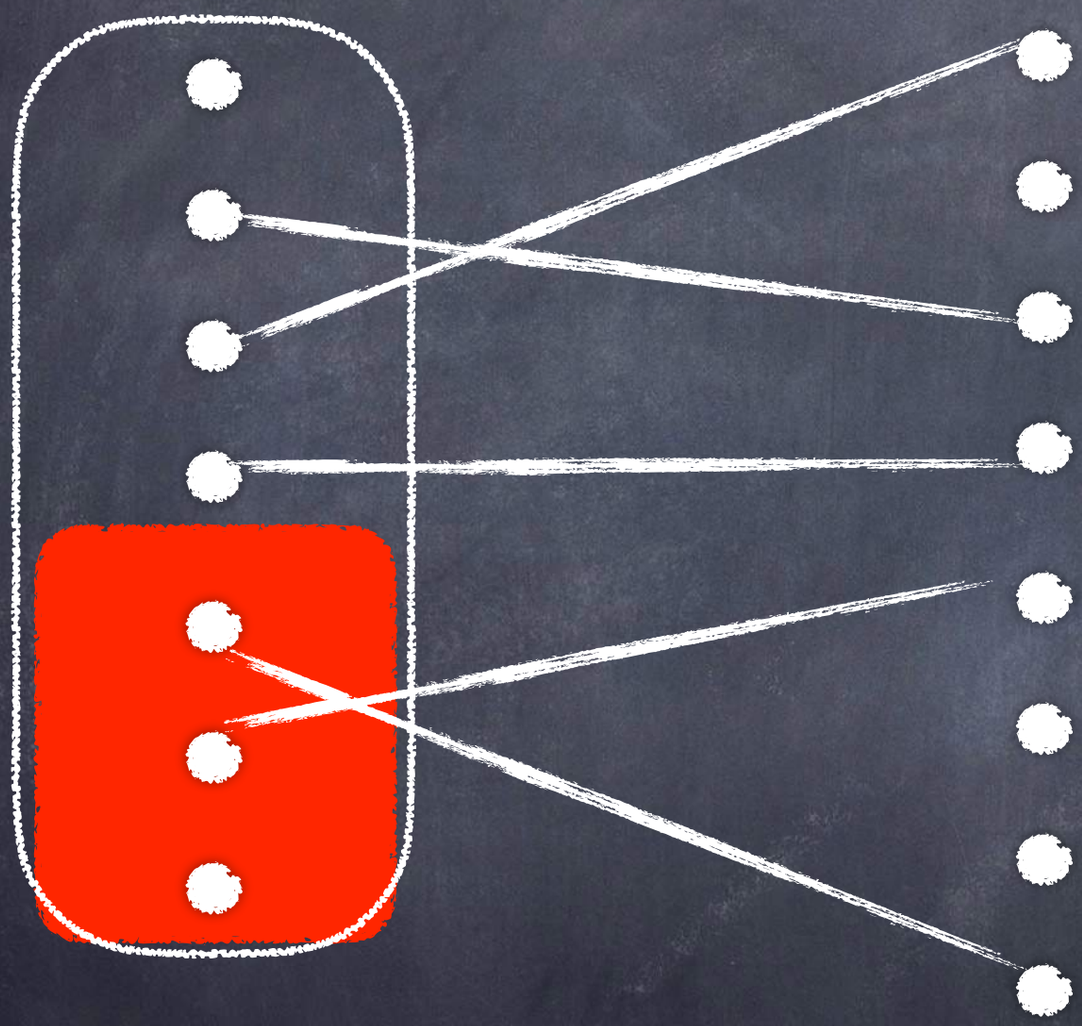
Compression

- Suppose that we found a (slightly) oversized solution
- Will it help to solve the problem?

Compression

- Suppose we have a vertex cover of size $k+1$
- How could it help to find a vertex cover of size k?

Compression



VC S of size $k+1$

Independent Set

Guess which part of S
will be in new VC and which not
- Branch into 2^{k+1} subproblems
Each of the subproblems is solvable
in polynomial time (Why?)

Running time 2^k poly(n)

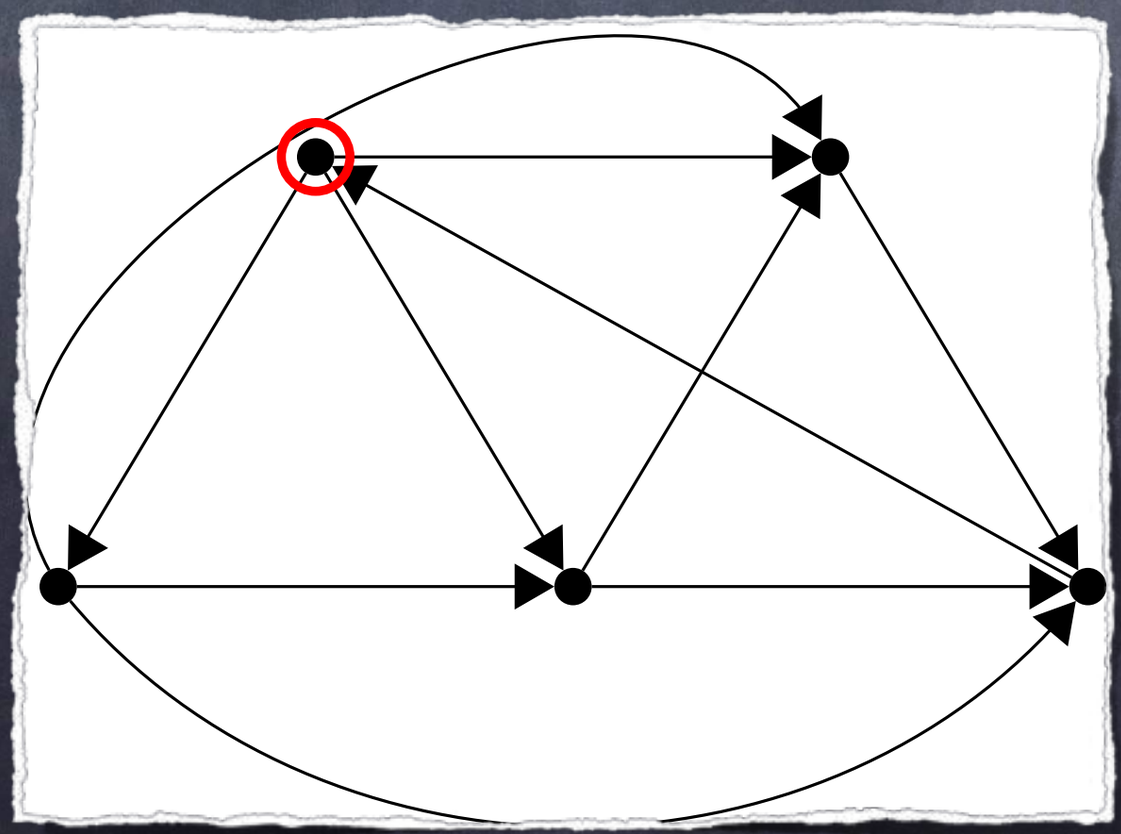
Iterative Step

- Let $v_1, v_2, \ldots, v_n$ be the vertices of G
- Start with a graph G_{k+1} induced by $v_1, v_2, \ldots, v_{k+1}$
- For graph G_p perform compression. If it has no VC of size k, reject. Otherwise, find a VC S of size k. In G_{p+1} set $S+v_{p+1}$ is a vertex cover of size $k+1$. Iterate.

Feedback Vertex Set in Tournament (FVST)

Input: A tournament (oriented clique) T and an integer k

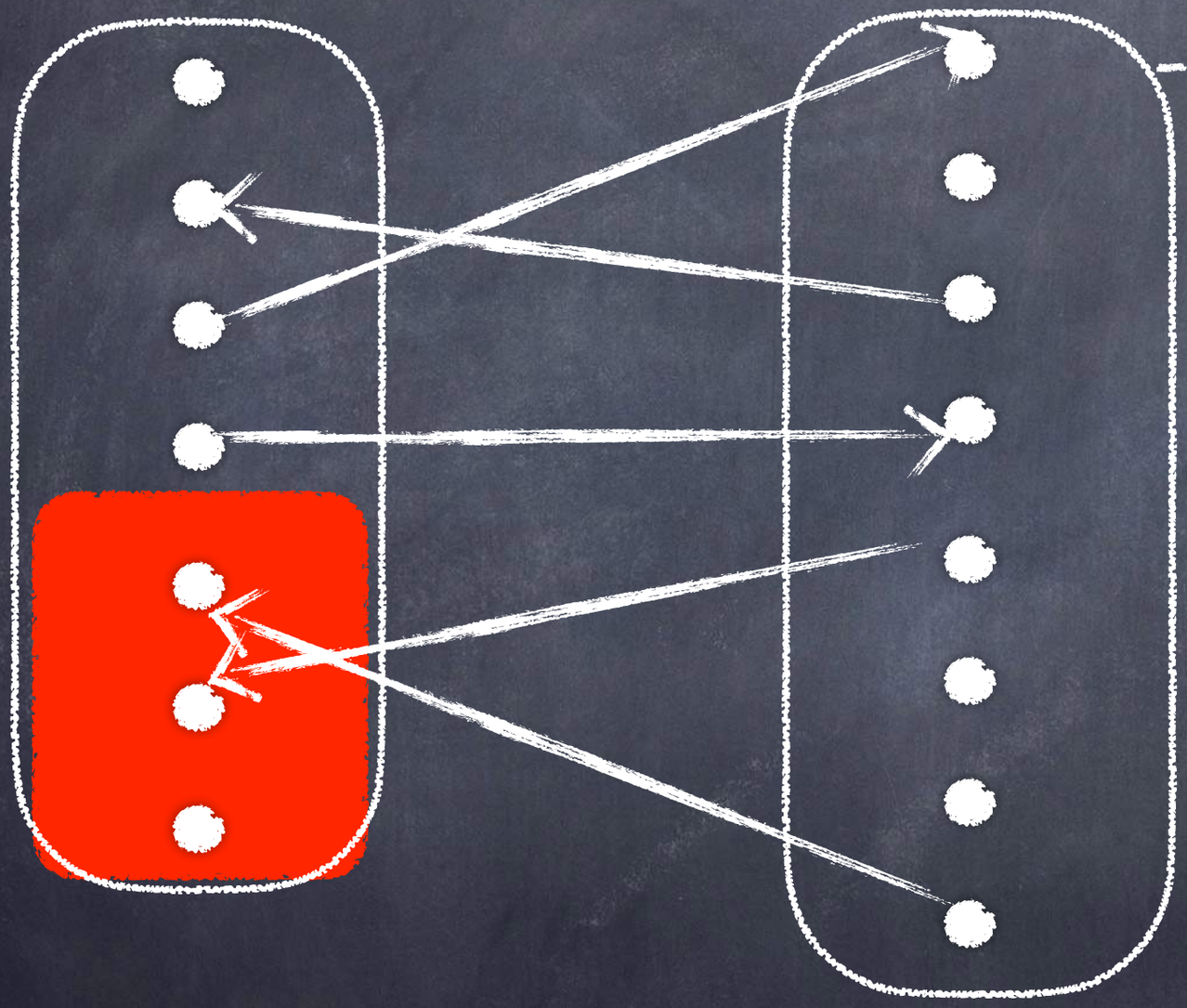
Question: Does there exist a subset X of $V(T)$ of size k such that $T-X$ is acyclic?



Compression

- Suppose we have a FVST X of size $k+1$
- How could it help to find a FVST of size k?

Compression



FVS X of size $k+1$

Acyclic Tournament

Guess which part of X
will be in new FVS and which not

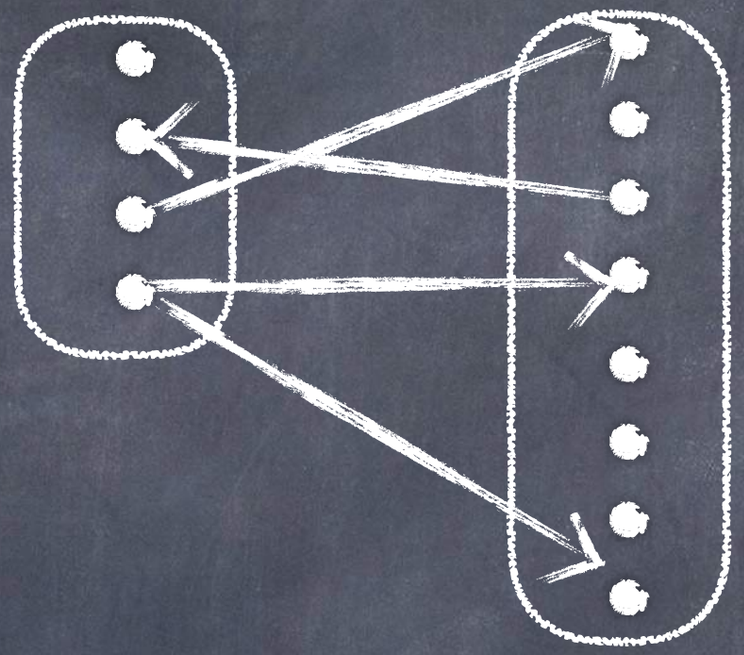
Branch into 2^{k+1} subproblems

Each of the subproblems is
solvable in polynomial time
(needs a proof)

Running time 2^k poly(n)

Disjoint FVST

FVS W of size p

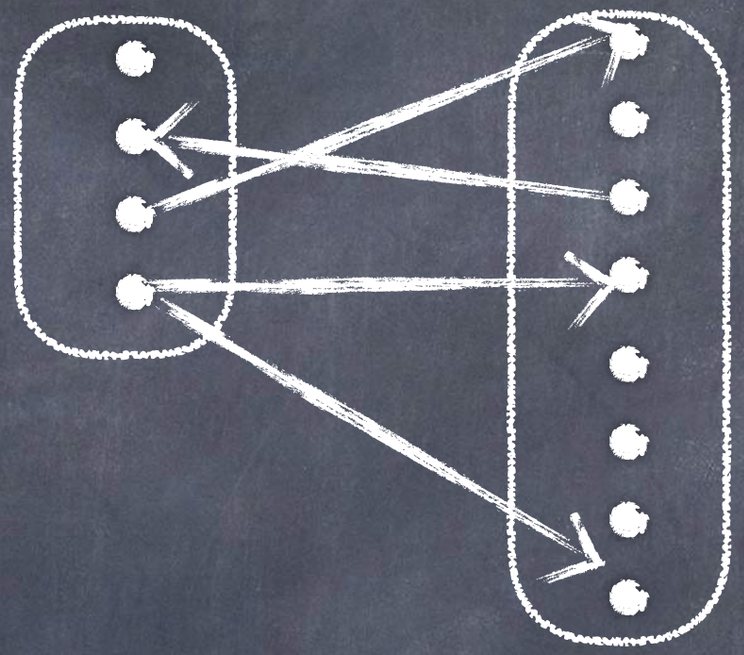


Acyclic Tournament A

INPUT: A tournament T with a FVS W of size p.
TASK: Decide, whether T has a FVS U of size $k-p$ disjoint with W

Disjoint FVST

FVS W of size p



Acyclic Tournament A

What is the maximum number of vertices X from A can be added to W to keep T[W+X] acyclic?

Lemma. Disjoint FVST is solvable in polynomial time

In other words: One can find the maximum number of vertices X from A such that $T[W+X]$ acyclic in polynomial time

Proof of Lemma.

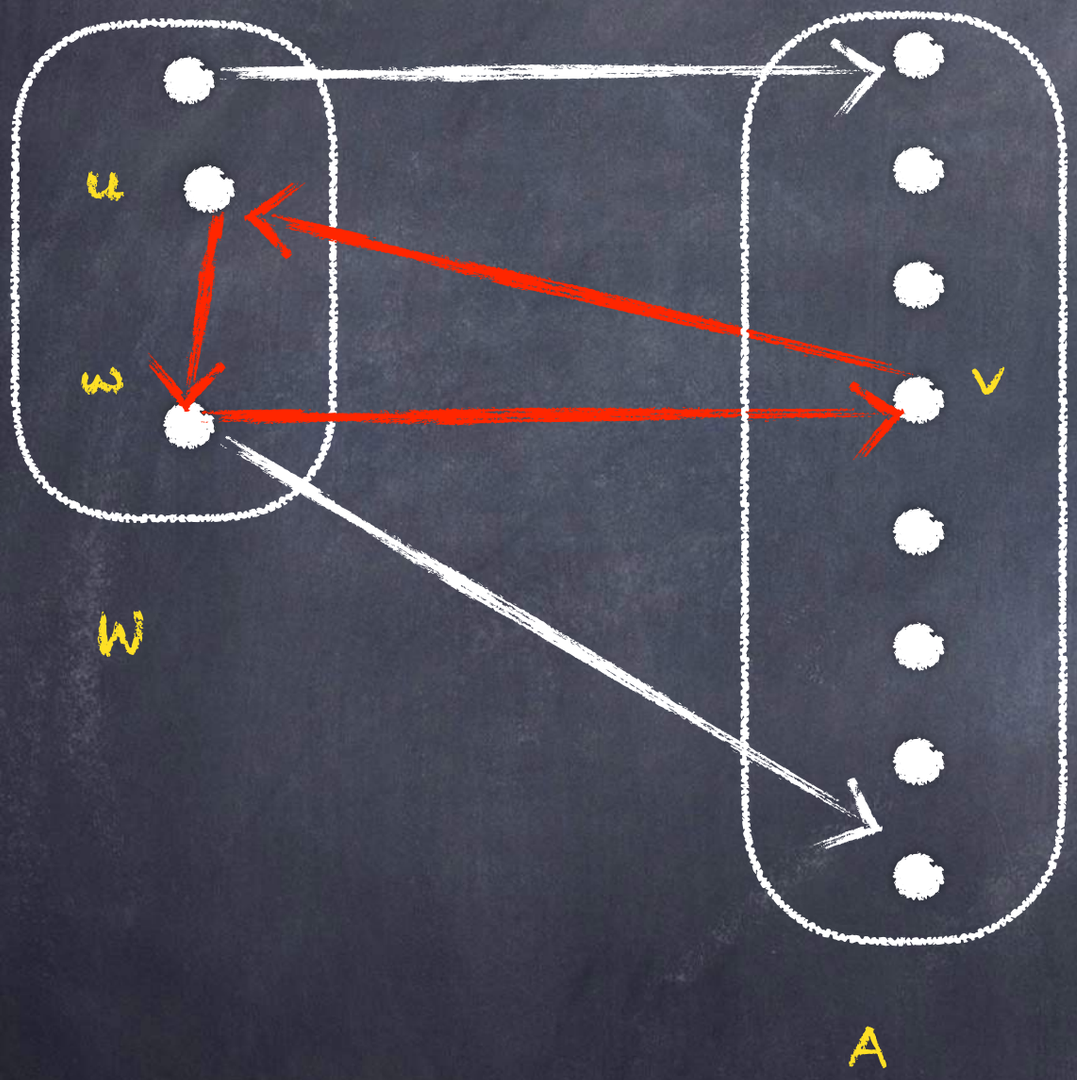
Fact:

A tournament is acyclic

Tournament does not contain a directed triangle

Tournament contains a (unique) topological ordereing

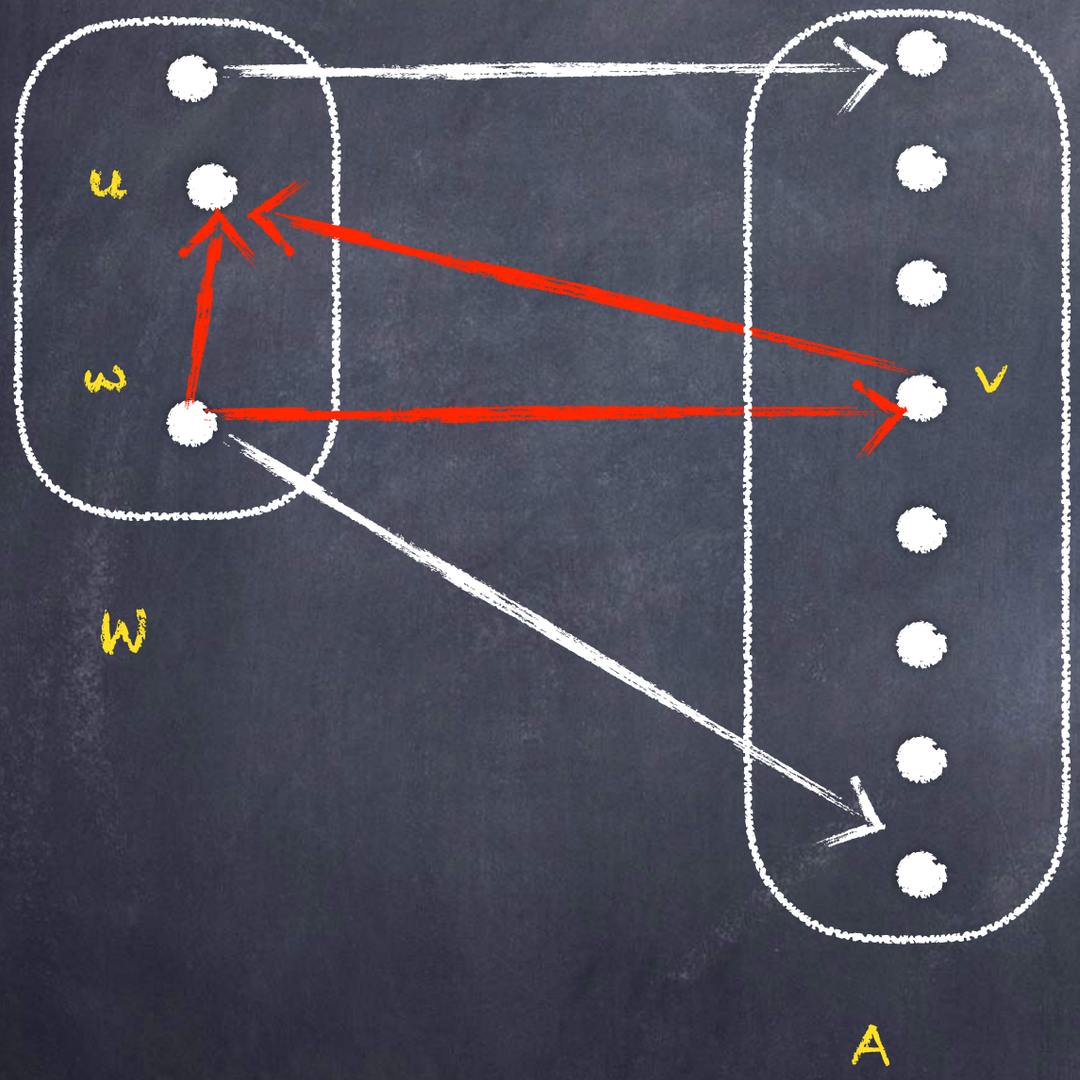
Proof of Lemma.



If there exist v in A and u, w in W s.t. vuw is a triangle, then v must be in FVS

Add v to W, decrease the parameter by 1

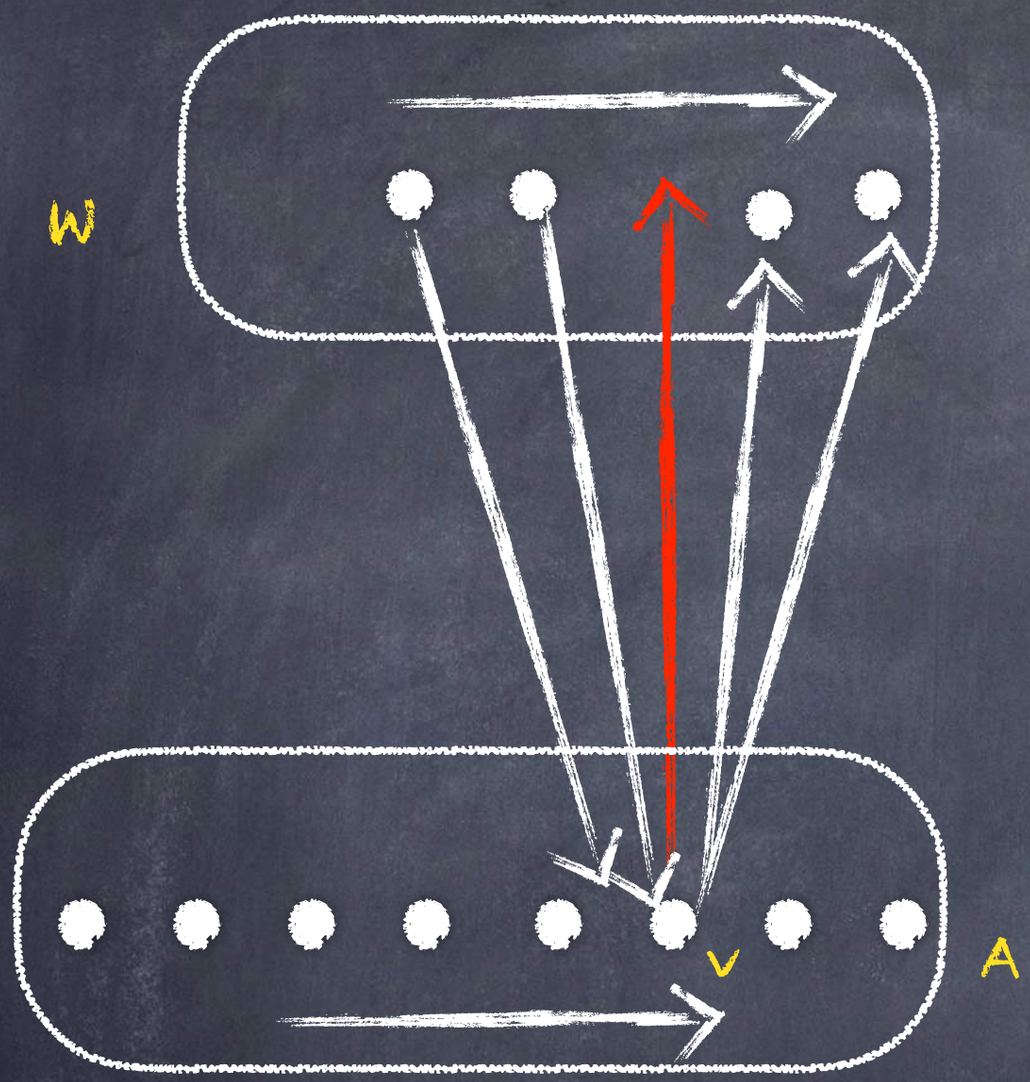
Proof of Lemma.



After transforming the tournament, for every v in A tournament T[W+v] is acyclic.

In topological ordering of tournament T[W+v] vertex v has its unique position.

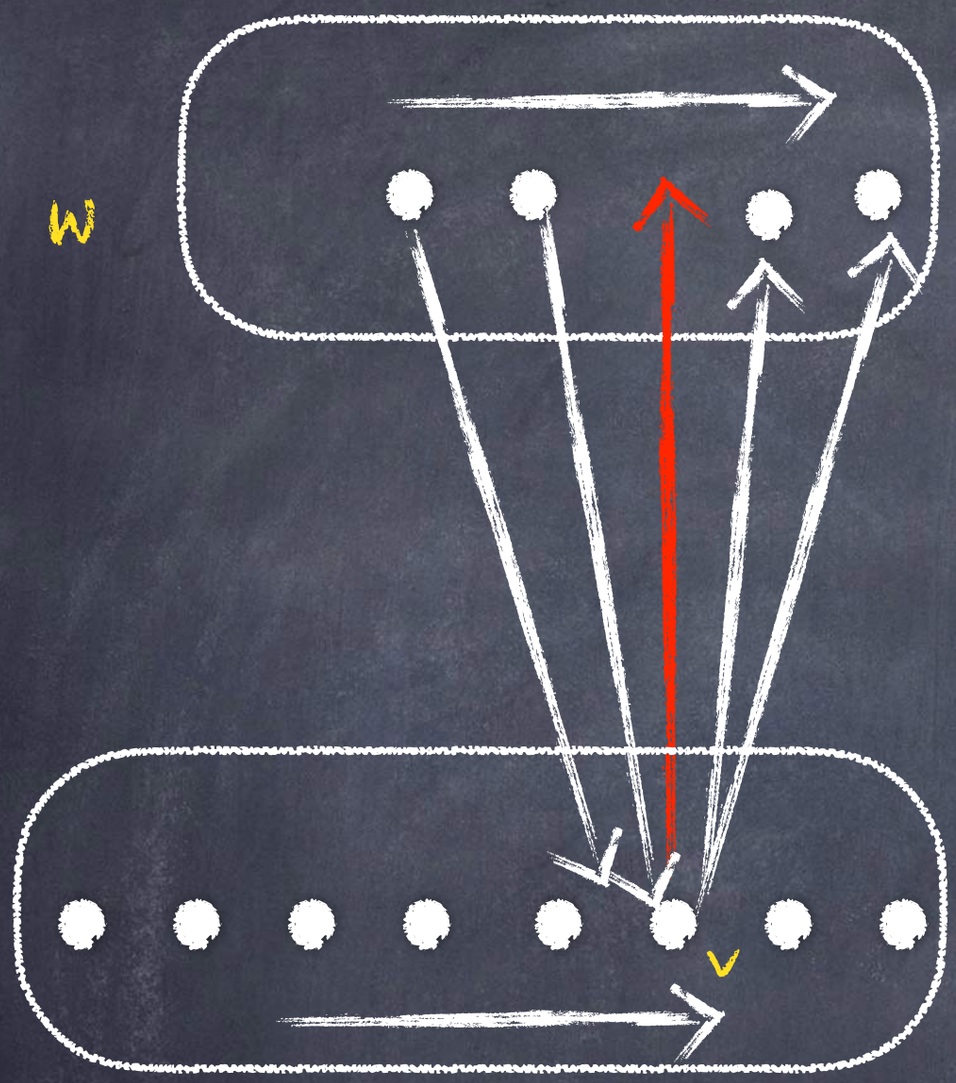
Proof of Lemma.



In topological ordering of tournament T[W+v] vertex v has its unique position.

This definies an ordering p of A

Proof of Lemma



This definies ordering p of A

p[u]<p[v]

- if position of u in t.o. of T[W+u] is smaller than position of v in t.o. of T[W+v],
- or u and v have equal positions but q[u]<q[v]

A

T[A] is acyclic, so A has a topological ordering q

FIND THE LONGEST COMMON SUBSEQUENCE OF p and q!!!

Lemma. Disjoint FVST is solvable in polynomial time

Compression of FVST is doable in 2^k poly(n) time

Theorem. Disjoint FVST is solvable in 2^k poly(n) time

Iterative Compression: Solving a problem (*)

- If there exist an algorithm solving (*)-COMPRESSION in time $f(k)\, n^c$ then there exists an algorithm solving problem (*) in time $f(k)\, n^{c+1}$
- If DISJOINT(*) is solvable in time $g(k)\, n^{O(1)}$ then (*)-COMPRESSION is solvable in time

$$\sum_{i=0}^{k} \binom{k+1}{i} g(k-i) n^{O(1)}$$

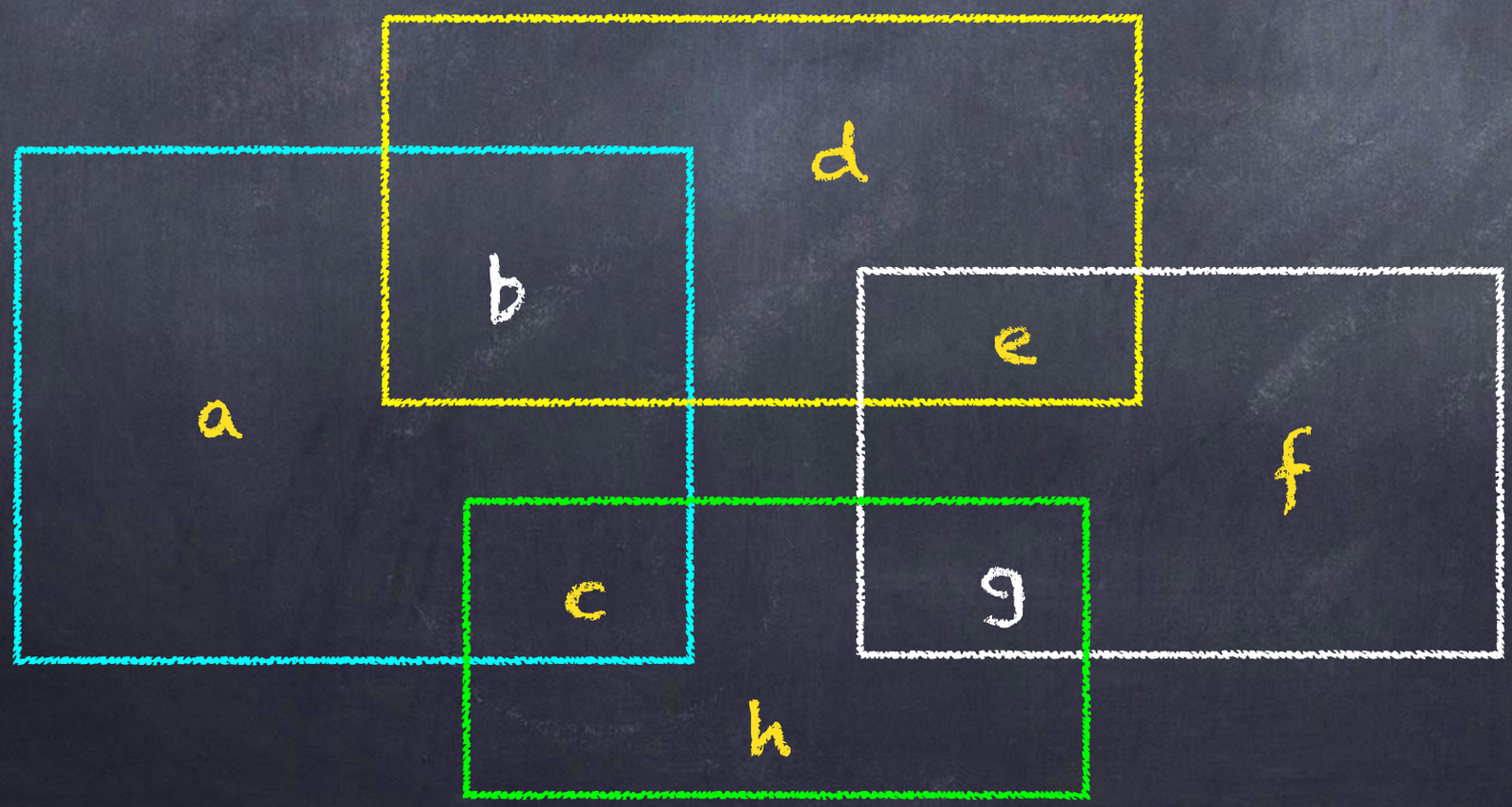
$$g(k) = \alpha^k \quad \Longrightarrow \quad (1+\alpha)^k n^{O(1)}$$

DISJOINT(*) → (*)-COMPRESSION

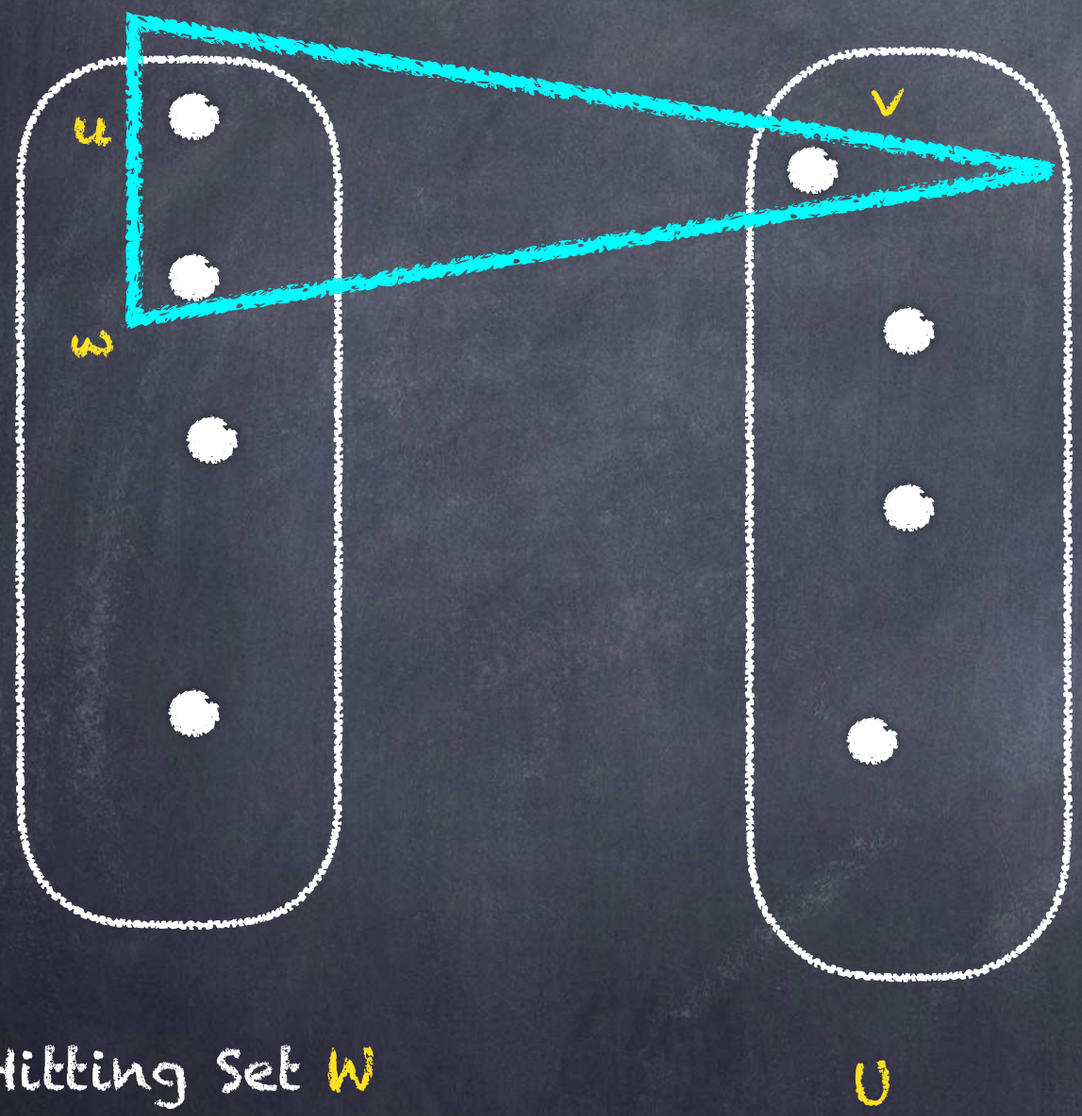
3-Hitting Set

Input: A universe U, a family A of sets of size 3 over U, integer k

Question: Does there exist a subset X of U of size k that has a nonempty intersection with every member of A?



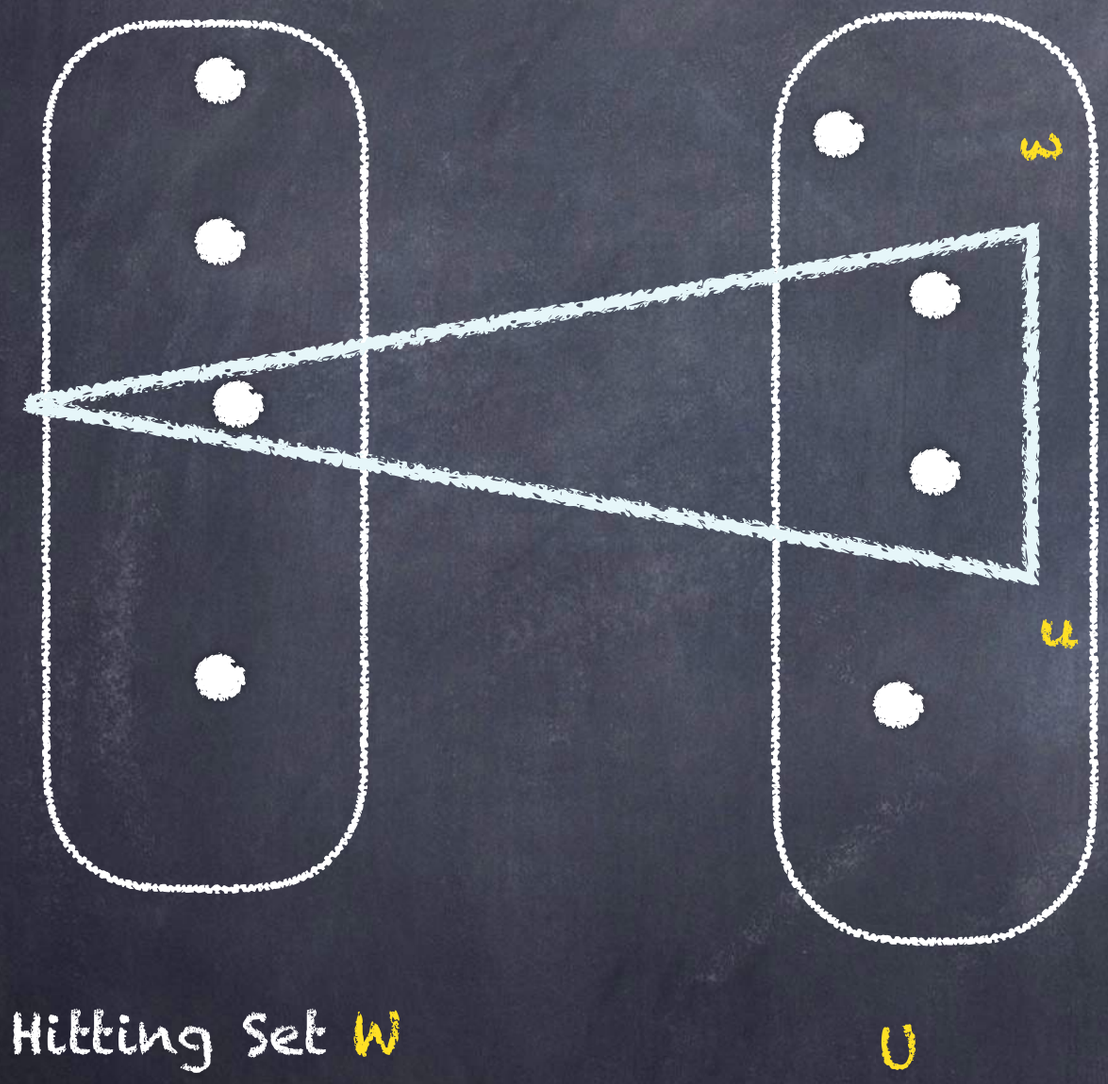
DISJOINT 3-Hitting Set



If there is set {v,u,w} s.t. v in U and u, w in W, then v must be in HS

Reduce k by 1 and delete from A all sets containing v

DISJOINT 3-Hitting Set



Now for every S={v,u,w} in A, v in W and u,w in U. Either u or w should be in HS

Build auxilary graph G_U on U. u and w are adjacent in G_U iff they are in a set of A

For every edge uw of G_U at least one of its endpoints should be in HS.

To solve

DISJOINT 3-Hitting Set

We solve

Vertex Cover

Reminder

- If there exist an algorithm solving (*)-COMPRESSION in time $f(k)\, n^c$ then there exists an algorithm solving problem (*) in time $f(k)\, n^{c+1}$
- If DISJOINT(*) is solvable in time $g(k)\, n^{O(1)}$ then (*)-COMPRESSION is solvable in time

$$\sum_{i=0}^{k} \binom{k+1}{i} g(k-i) n^{O(1)}$$

$g(k) = \alpha^k$ ⟹ $(1+\alpha)^k n^{O(1)}$

DISJOINT(*) ⟹ (*)-COMPRESSION

Theorem. Let c be a constant such that VC is solvable in time $c^k \cdot \text{poly}(n)$. Then 3-HS is solvable in time $(1+c)^k \cdot \text{poly}(n)$

More problems to iterate...

Exercises

- Cluster Vertex Deletion: 2^k poly(n)
- Feedback Vertex Set (undirected graph): $2^{O(k)}$ poly(n)
- Odd Cycle Transversal: $2^{O(k)}$ poly(n)
- d-HS in time $(1+c)^k$ poly(n), where c is the constant for (d-1)-HS

Summing up

Two basic techniques for FPT algorithms

- Branching (first thing to try)
  - Vertex Cover, 3-Hitting Set, Feedback Vertex Set in Tournaments
- Iterative Compression
  - VC, FVST, 3-Hitting Set

Reading: chapters 3 and 4 of PA book