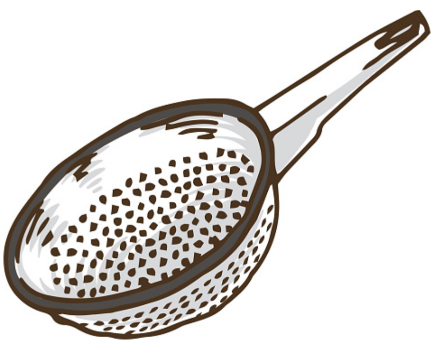
# Faster FPT algorithms using algebraic methods

- Inclusion-exclusion principle
  - Counting Hamiltonian paths in time $O^*(2^n)$ and polynomial space.
  - Steiner tree in time $O^*(2^k)$ and polynomial space.
- Polynomials
  - k-path in time $O^*(2^k)$ and polynomial space.

**Sieving** : Sieve out unwanted objects using algebraic cancellations.



Lecture #8

December 07, 2021

Roohani Sharma

## Counting HAMILTONIAN PATH problem

Input: A directed graph G
Output: The number of Hamiltonian paths in G?

Recall from Exercise sheet #2:
Hamiltonian path can be solved in time $O^*(2^n) = O(2^n n^{O(1)})$
(using dynamic programming over subsets).

Proof idea:
For every vertex subset X,
T[X,v] =1, if there exists a path starting at v with vertex set X.

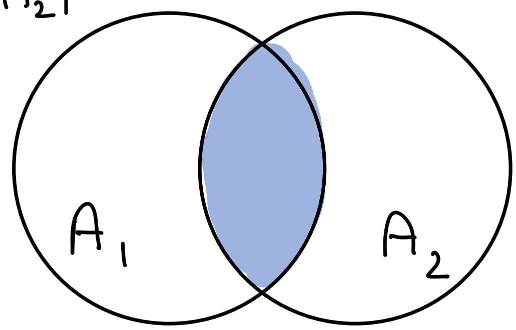
Space complexity: $O^*(2^n)$.

**Goal**: Counting Hamiltonian path in time $O^*(2^n)$
and polynomial space
using the inclusion-exclusion principle.

# Inclusion-exclusion principle

Given: Sets $A_1, A_2$

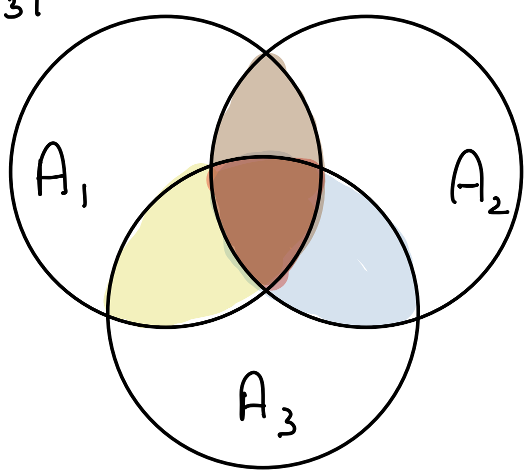
Compute: $|A_1 \cup A_2|$




$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

Given: Sets $A_1, A_2, A_3$

Compute: $|A_1 \cup A_2 \cup A_3|$




$$|A_1 \cup A_2 \cup A_3| = |A_1| + |A_2| + |A_3|$$

$$- |A_1 \cap A_2| - |A_1 \cap A_3| - |A_2 \cap A_3|$$

$$+ |A_1 \cap A_2 \cap A_3|$$

# Inclusion-exclusion principle (union version)

Let $A_1, A_2, \ldots, A_n \subseteq U$; $U$ is a finite set (Universe).

Then, $[n] = \{1, 2, \ldots, n\}$

$$\left| \bigcup_{i \in [n]} A_i \right| = \sum_{\phi \neq X \subseteq [n]} (-1)^{|X|+1} \left| \bigcap_{i \in X} A_i \right|$$

Useful: When finding intersection of sets is easier than finding their union.

Proof:

Fix $e \in \bigcup_{i \in [n]} A_i$. Show that $e$ is counted exactly once in R.H.S.

Let $e$ be present in exactly the sets $A_{i_1}, A_{i_2}, \ldots, A_{i_k}$.

Then,

$$e \in \bigcap_{i \in X} A_i \quad \text{iff} \quad X \subseteq \{i_1, i_2, \ldots, i_k\}$$

Contribution of $e$ in RHS $= \sum_{\phi \neq X \subseteq \{i_1, \ldots, i_k\}} (-1)^{|X|+1}$

$$= (-1) \sum_{j=1}^{k} \binom{k}{j} (-1)^j = (-1)\left((-1+1)^k - 1\right) = 1.$$

Use Binomial theorem: $(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} a^i b^{n-i} = b^n + \sum_{i=1}^{n} \binom{n}{i} a^i b^{n-i}$

## Inclusion-exclusion principle (intersection version)

Let $A_1, A_2, \ldots, A_n \subseteq U$; $U$ is a finite set (Universe).

Then

$$\left|\bigcap_{i\in[n]} A_i\right| = \sum_{X\subseteq[n]} (-1)^{|X|} \left|\bigcap_{i\in X} \overline{A_i}\right|$$

$\overline{A_i} = U \setminus A_i$.

Useful: When finding the union of the sets is easier than finding their intersection.

Proof hint: Use De Morgan's law on the formula for inclusion-exclusion principle (union version).

De Morgan's law:
$$\overline{A_1 \cup A_2} = \overline{A_1} \cap \overline{A_2}$$

$$\overline{A_1 \cap A_2} = \overline{A_1} \cup \overline{A_2}$$

# Inclusion-exclusion principle (intersection version)

Let $A_1, A_2, \ldots, A_n \subseteq U$;
$U$ is a finite set (Universe).
Then

$$\left|\bigcap_{i \in [n]} A_i\right| = \sum_{X \subseteq [n]} (-1)^{|X|} \left|\bigcap_{i \in X} \overline{A}_i\right|$$

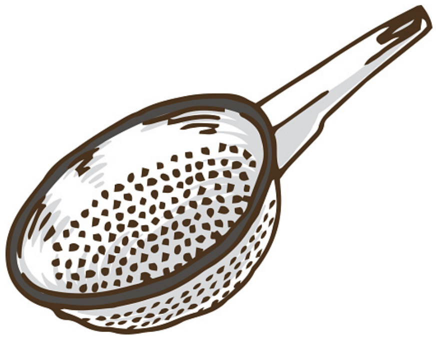
$\overline{A}_i = U \setminus A_i$.

Useful: When finding the union of the sets is easier than finding their intersection.

* If $\bigcap_{i \in X} \overline{A}_i$ is computable in poly time $\forall X$,
then $\bigcap_{i \in [n]} A_i$ is computable via $O^*(2^n)$ arithmetic operat
and in polynomial space.

Applications →

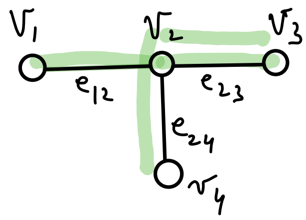
# Counting Hamiltonian path problem



## Relaxing paths - walks

**Walk**: is a traversal of the edges of the graph with the possibility of repeating vertices.

Ex:




$(v_1\ e_{12}\ v_2\ e_{23}\ v_3\ e_{23}\ v_2\ e_{24}\ v_4)$ is a 4-length walk from $v_1$ to $v_4$.

**Length** of a walk is the number of edges it uses.
Here repetitions are counted each time.
**Every k-path is a k-walk**.

k-walk is a walk of length k.
A walk from vi to vj is a walk that starts at vi and ends at vj

# Counting k-WALKS is polynomial time solvable

Consider an undirected graph G.
(also works for directed graphs- Exercise)

Let M be the adjacent matrix of G.

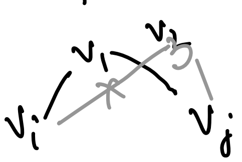
$$M = \begin{array}{c c} & \begin{array}{ccccc} v_1 & \cdots & v_j & \cdots & v_n \end{array} \\ \begin{array}{c} v_1 \\ \vdots \\ v_i \\ \vdots \\ v_n \end{array} & \begin{bmatrix} & & & & \\ & & & & \\ & & m_{ij} & & \\ & & & & \\ & & & & \end{bmatrix} \end{array}$$

$m_{ij} = 1$ iff $(v_i, v_j) \in E(G)$

$m_{ij} = \#$ of 1-walks from $v_i$ to $v_j$.

Fix $i$, $\sum_{j=1}^{n} m_{ij} = \#$ of 1-walks starting at $v_i$.

$$M^2 = \begin{array}{c c} & \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ v_i & \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array} \begin{array}{c c} & v_j \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} & \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \end{array}$$




$$= \begin{array}{c c} & \begin{array}{ccccc} v_1 & \cdots & v_j & \cdots & v_n \end{array} \\ \begin{array}{c} v_1 \\ \vdots \\ v_i \\ \vdots \\ v_n \end{array} & \begin{bmatrix} & & & & \\ & & & & \\ & & m^2_{ij} & & \\ & & & & \\ & & & & \end{bmatrix} \end{array}$$

$m^2_{ij} = \#$ of 2-walks from $v_i$ to $v_j$

$\sum_{j=1}^{n} m^2_{ij} = \#$ of 2-walks starting at $v_i$.

# Counting k-WALKS is polynomial time solvable (contd.)

$$M^K = M^{K-1} \cdot M = \begin{array}{cc} & \\ v_i & \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \end{array} \begin{array}{c} \quad v_j \\ \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \end{array}$$

$$= \begin{array}{cc} & \begin{array}{ccccc} v_1 & \cdots & v_j & \cdots & v_n \end{array} \\ \begin{array}{c} v_1 \\ \vdots \\ v_i \\ \vdots \\ v_n \end{array} & \begin{bmatrix} & & & & \\ & & & & \\ & & m_{ij}^k & & \\ & & & & \\ & & & & \end{bmatrix} \end{array}$$

$m_{ij}^k$ = # of k-walks from $v_i$ to $v_j$

$\sum_{j=1}^{n} m_{ij}^k$ = # of k-walks starting at $v_i$.

(Exercise: use induction).

$$\boxed{\#\text{ k-walks in } G = \sum_{i=1}^{n} \sum_{j=1}^{n} m_{ij}^k}$$

Since matrix multiplication can be done in polynomial time, counting k-walks can be done in polynomial time.

(Indeed counting k-walks is easier!)

# Back to counting Hamiltonian paths

- use inclusion-exclusion for sieving.

$$\left|\bigcap_{v\in V(G)} A_v\right| = \sum_{X\subseteq V(G)} (-1)^{|X|} \left|\bigcap_{v\in X} \bar{A}_v\right|$$

Let $U$ = set of all $(n-1)$-walks of $G$. $(n = |V(G)|)$

$\forall v \in V(G)$,

$A_v$ = set of all $(n-1)$-walks containing $v$.

Observe, $\left|\bigcap_{v\in V(G)} A_v\right|$ = # of Hamiltonian paths.

(set of all $(n-1)$-walks that contain $V(G)$).

Claim: For any $X \subseteq V(G)$, $\left|\bigcap_{v\in X} \bar{A}_v\right|$ can be computed in polynomial time.

Proof:

$\left|\bigcap_{v\in X} \bar{A}_v\right|$ = set of all $n$-walks not containing any vertex of $X$

= set of all $n$-walks in $G-X$

$\blacksquare$

* $\left|\bigcap_{v \in X} \bar{A}_v\right|$ can be computed in polynomial time.
* $\left|\bigcap_{v \in X} \bar{A}_v\right| \leq n^n \Rightarrow \left|\bigcap_{v \in X} \bar{A}_v\right|$ can be represented using $O(n \log n)$ bits.

\+ Inclusion-exclusion

$\Downarrow$

$\left|\bigcap_{v \in V(G)} A_v\right|$ can be computed using $O^*(2^n)$ arithmetic operations of $O(n \log n)$ bit numbers,

in polynomial space.

Theorem:

$$\boxed{\text{Counting Hamiltonian paths can be done in } O^*(2^n) \text{ time and polynomial space.}}$$

Application 2 $\longrightarrow$

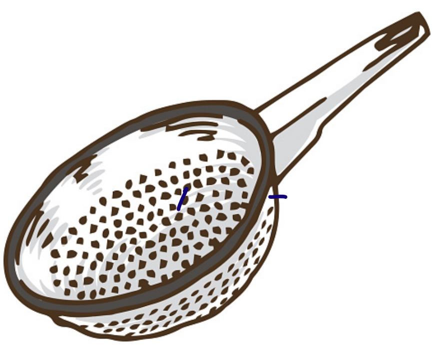
# STEINER TREE problem

Recall from lecture #3:

**STEINER TREE**
Input: A graph G, a set of terminals K, an integer p
Question: Does there exist a tree with ≤ p edges that contain K?

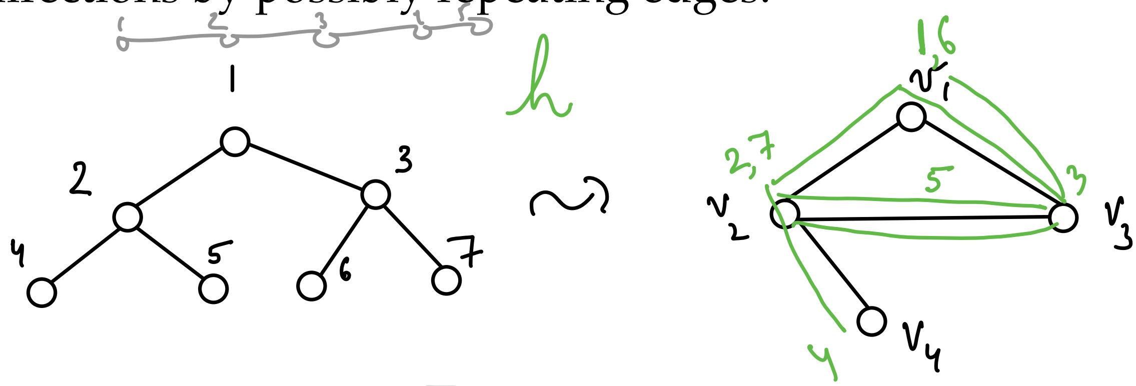
Steiner tree can be solved in time $O^*(3^p)$
using DP over subsets.

**Goal**: Steiner tree can be solved in time $O^*(2^p)$
and polynomial space,
using inclusion-exclusion.



# Relaxing trees — branching walks

**Branching walk:** traverse the edges in possibly multiple directions by possibly repeating edges.




Ordered, rooted tree H

**G**

**Homomorphism** H to G: $h : V(H) \to V(G)$ such that if $(x,y) \in E(H)$, then $(h(x), h(y)) \in E(G)$.

(homomorphisms preserve edges)

> A branching walk (H,h) is a homomorphism h from an ordered, rooted tree H to G.

The **length** of a branching walk (H,h) is $|E(H)|$.

A **k-branching walk** is a branching walk of length k.

A branching walk (H,h) is a tree if and only if h is injective.

A branching walk (H,h) starts at a vertex v of G if h(root)=v. It contains a vertex v of G if there exists some x in V(H) such that h(x)=v.

# Computing #of k-branching walks in polynomial time

Use dynamic programming.

$\forall\, x \in V(G)$ and $j \in \{0, \dots, p\}$,

$T_j[x]$ = # of $j$-branching walks starting at $v$.

(We are looking for $\sum_{x \in V(G)} T_k[x]$.)

Computing $T_j[x]$:

$$T_j[x] = \begin{cases} 1 & \text{if } j=0 \\ \displaystyle\sum_{y \in N(x)} \sum_{j_1+j_2=j-1} T_{j_1}[x] \cdot T_{j_2}[y] & \text{otherwise} \end{cases}$$


Exercise: Prove the formal correctness.

Computing #of p-branching walks in any graph can be done in polynomial time.

# Back to the Steiner tree problem

$\mathcal{U}$ = set of all $p$-branching walks

$\forall v \in K$,
$A_v$ = set of all $p$-branching walks contains at $v$.

$\left| \bigcap_{v \in K} A_v \right|$ = # of $p$-branching walks that contain all vertices of $K$

Check (as an exercise) that $\left| \bigcap_{v \in K} A_v \right|$ does not count all Steiner trees for $K$ with $p$ edges.

Claim: $\left| \bigcap_{v \in K} A_v \right| \neq 0$ iff $\exists$ a Steiner tree for $K$ with $p$ edges.

<u>Proof:</u>

$\Leftarrow$

$\Rightarrow$

$\blacksquare$

$\therefore$ Enough to check if $\left| \bigcap_{v \in K} A_v \right| = 0$.

Using inclusion-exclusion, it is enough to compute,

$$\left| \bigcap_{X \subseteq K} \overline{A}_v \right| , \quad \text{for each } X \subseteq K.$$

* $\left| \bigcap_{X \subseteq K} \overline{A}_v \right| = \#$ of $p$-branching walks in $G-X$.

$\therefore$ can be found in polynomial time.

* $\left| \bigcap_{X \subseteq K} \overline{A}_v \right| = O(n^p)$

$\therefore$ can be encoded by $O(p \log n)$ bits.

Theorem:

$\boxed{\text{Steiner Tree can be solved in } O^*(2^p) \text{ time and polynomial space.}}$

# Polynomials

$$p(x) = x+1$$

or

$$= x^2 + x - 6$$

$$p(x_1, x_2) = x_1^2 + x_2^2 + 2x_1x_2$$

or

$$= x_1^2 + 3x_1x_2 + 10x_2 - 40$$

$$p(x_1, \dots, x_n) = \sum_{(c_1, \dots, c_n) \in (\mathbb{N} \cup \{0\})^n} \underbrace{\boxed{a_{c_1, \dots, c_n} \, x_1^{c_1} x_2^{c_2} \cdots x_n^{c_n}}}_{\text{monomial}}$$

where $a_{c_1, \dots, c_n} \in$ ? field F

and $a_{c_1, \dots, c_n}$ are non-zero only for a finite number of tuples $(c_1, \dots, c_n)$.

degree of $p(\cdot) = \max\limits_{a_{c_1, \dots, c_n}} (c_1 + c_2 + \dots + c_n)$

$p(x_1, \dots, x_n)$ interpreted over field $(F, +, \cdot)$.

# Field (F,+,.)

A field is a triple (F,+,.) such that the following hold:

- F is a set
  $+$ and $\cdot$ are binary operations
- $\forall\, a, b \in F$, $a+b \in F$, $a.b \in F$
- Associativity: $(a+b)+c = a+(b+c)$,
  $(a.b).c = a.(b.c)$
- Commutativity: $a+b = b+a$, $a.b = b.a$
- Distributivity: $a\cdot(b+c) = a.b + a.c$
- Additive Identity: $\exists\, 0 \in F$ such that $\forall a \in F$, $0+a = a$
- Multiplicative Identity: $\exists\, 1 \in F$ such that $\forall a \in F \setminus \{0\}$, $1.a = a$
- Additive Inverses: $\forall a \in F$, $\exists\, b \in F$ such that $a+b=0$
- Multiplicative Inverses: $\forall a \in F \setminus \{0\}$, $\exists\, b \in F$ such that $a.b = 1$

Examples: $(\mathbb{Q}, +, \times)$, $(\mathbb{R}, +, \times)$, $(\mathbb{C}, +, \times)$
These are examples of infinite fields.

We are interested in finite fields.

# Finite fields

For every integer $s \geq 1$ and a prime $p$, there exists a finite field of size $p^s$.

- When $s = 1$,
  $F_p = (\{0, 1, \dots, p-1\}, + \bmod p, \times \bmod p)$
  is a finite field.
- When $s > 1$, it is non-trivial to construct these fields (beyond the scope of the course).

We will be interested in finite fields of size $2^s$. For the purposes of the course it is enough to remember the following facts (without proof)

## About finite fields of size $2^s$

(1) They exist for every $s \geq 1$.
(2) Arithmetic operations can be performed fast, in time $O(s \log s \log \log s)$.
(3) They are of characteristic two, that is, $1+1=0$.
In particular, for any element $a$ of this field,
$a+a = a.(1+1) = a.0 = 0$.

## k-PATH problem

Input: A (directed) graph, an integer k
Question: Does there exist a path on k vertices (k-path) in G?

Recall from lecture #2:

k-path can be solved in time $O^*((2e)^k)$
using Color Coding.

**Goal**: k-path can be solved in time $O^*(2^k)$
by a randomized algorithm,
using polynomial identity testing (PIT).
This can be done in polynomial space
by additional using the inclusion-exclusion principle.

PIT: Is $p(x_1, \ldots, x_n)$ identically zero,
that is,
is $p(x_1, \ldots, x_n) = 0$
for all $(x_1, \ldots, x_n)$ ?

Ex: $p(x) = \cancel{x^2} - \cancel{2x^2} + \cancel{x} + \cancel{x^2} - \cancel{x} = 0$
is identically zero.

## k-PATH problem

Input: A (directed) graph, an integer k
Question: Does there exist a path on k vertices (k-path) in G?

Algorithm for k-path problem

Step 1: Reduce k-path to PIT.
Step 2: Solve the instance of PIT (Randomized).

# Reduce k-PATH to PIT

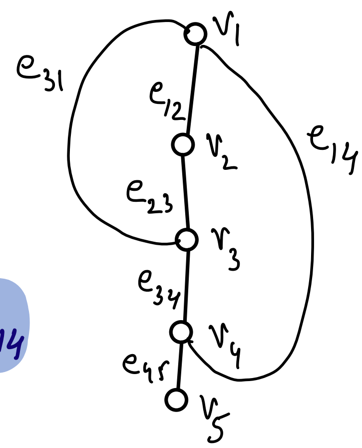
Goal: Construct a polynomial $p(\cdot)$ such that G has a k-path iff $p(\cdot) \neq 0$.

## Constructing the polynomial p

Attempt 1: Construct a monomial for every k-walk.

For every vertex $v$, create a variable $x_v$.

For every edge $e=(u,v)$, create a variable $y_{uv}$.




Walk W: $v_1\, e_{12}\, v_2\, e_{23}\, v_3\, e_{31}\, v_1\, e_{14}\, v_4$

$x_{v_1}$ $x_{v_1}$

$$mon_W = x_{v_1}^2 x_{v_2} x_{v_3} x_{v_4} y_{12} y_{23} y_{31} y_{14}$$

Ex: Path W': $v_1\, e_{12}\, v_2\, e_{23}\, v_3\, e_{34}\, v_4\, e_{45}\, v_5$

$$mon_{W'} = x_{v_1} x_{v_2} x_{v_3} x_{v_4} x_{v_5} y_{12} y_{23} y_{34} y_{45}$$ (Multilinear)

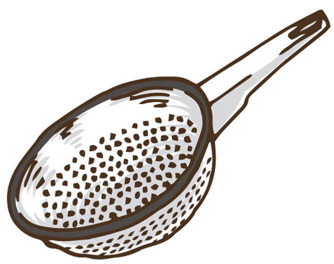
Attempt 1:

$$p(\mathbf{x},\mathbf{y}) = \sum_{\substack{\text{all} \\ k\text{-walks} \\ W}} \text{mon}_W(\mathbf{x},\mathbf{y})$$

If $p(\mathbf{x},\mathbf{y}) \neq 0$ for all $(\mathbf{x},\mathbf{y})$,

then is there a $k$-path in $G$?

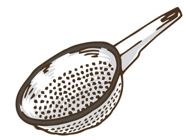
– Not necessarily.



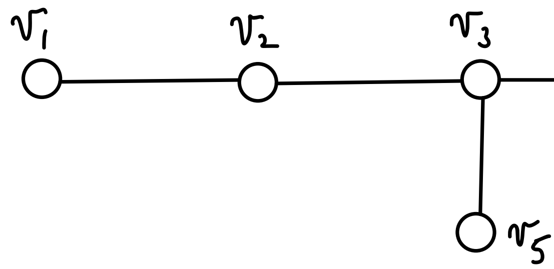
**Non-path walks should cancel out!**

Recall $p(\cdot)$ will be interpreted over $(\mathbb{F}_{2^s}, +, \times)$ which is a field of characteristic 2.

Ex. $x_{v_1}^2 x_{v_2} x_{v_3} y_{12} y_{23} y_{31} + x_{v_1}^2 x_{v_2} x_{v_3} y_{12} y_{23} y_{31} = 0$.



## Attempt 2 (Final): Introduce labelled walks.




## Labelled walk

$(W, l_1)$: $v_1 \; e_{12} \; v_2 \; e_{23} \; v_3 \; e_{35} \; v_5 \; e_{35} \; v_3 \; e_{34} \; v_4$

$(W, l_2)$: $v_1 \; e_{12} \; v_2 \; e_{23} \; v_3 \; e_{35} \; v_5 \; e_{35} \; v_3 \; e_{34} \; v_4$

$(W, l_3)$: $v_1 \; e_{12} \; v_2 \; e_{23} \; v_3 \; e_{35} \; v_5 \; e_{35} \; v_3 \; e_{34} \; v_4$

⋮

**Labelled walk** is a pair $(W, l)$,
where $W = v_1, \dots, v_k$ is a $k$-walk
and $l : [k] \to [k]$ is a bijection.

# Construct a monomial for every labelled walk $(W, \ell)$.

For every vertex $v$ and $i \in [k]$ $\left(i \in \{\cdot, \cdot, \cdot, \cdot, \cdot\}\right)$, create a variable $x_{v,i}$ $\left( x_{v,\cdot} \right)$.

For every edge $e = (u, v)$, create a variable $y_{u,v}$.

$(W, \ell)$: $v_1 \;\; e_{12} \;\; v_2 \;\; e_{23} \;\; v_3 \;\; e_{35} \;\; v_5 \;\; e_{35} \;\; v_3 \;\; e_{34} \;\; v_4$

$$mon_{(W,\ell)} = x_{v_1,\cdot}\, x_{v_2,\cdot}\, x_{v_3,\cdot}\, x_{v_5,\cdot}\, x_{v_3,\cdot}\, x_{v_4,\cdot}\; y_{1,2}\, y_{2,3}\, y_{3,5}^2\, y_{3,4}$$

$$p(\mathbf{x}, \mathbf{y}) = \sum_{\substack{\text{all labelled} \\ k\text{-walks } (W,\ell)}} mon_{(W,\ell)}(\mathbf{x}, \mathbf{y})$$

$$= \sum_{k\text{-walk } W} \;\; \sum_{\substack{\ell : [k] \to [k] \\ \ell \text{ is a bijection}}} mon_{(W,\ell)}(\mathbf{x}, \mathbf{y})$$

## Claim:

$$p(\mathbf{x},\mathbf{y}) = \sum_{\substack{\text{all labelled} \\ k\text{-paths } W}} \mathrm{mon}_{(W,\ell)}(\mathbf{x},\mathbf{y})$$

**Proof Sketch:** Show that monomials for labelled walks that are not paths, cancel out.

Fix a walk $W$,

$(W,\ell_1) = v_1 \; e_{12} \; v_2 \; e_{23} \; v_3 \; e_{35} \; v_5 \; e_{35} \; v_3 \; e_{34} \; v_4$

$\mathrm{mon}_{(W,\ell_1)} = x_{v_1,\bullet}\, x_{v_2,\bullet}\, x_{v_3,\bullet}\, x_{v_5,\bullet}\, x_{v_3,\bullet}\, x_{v_4,\bullet}\; y_{1,2}\, y_{2,3}\, y_{3,5}^2\, y_{34}$

$(W,\ell_2) = v_1 \; e_{12} \; v_2 \; e_{23} \; v_3 \; e_{35} \; v_5 \; e_{35} \; v_3 \; e_{34} \; v_4$

$\mathrm{mon}_{(W,\ell_2)} = x_{v_1,\bullet}\, x_{v_2,\bullet}\, x_{v_3,\bullet}\, x_{v_5,\bullet}\, x_{v_3,\bullet}\, x_{v_4,\bullet}\; y_{1,2}\, y_{2,3}\, y_{3,5}^2\, y_{3,4}$

Observe: $\mathrm{mon}_{(W,\ell_1)} = \mathrm{mon}_{(W_2,\ell)}$

$\blacksquare$

**Observe:** Monomials for labelled paths do not cancel out.

(1) Distinct paths will have distinct monomials (irrespective of the labelling function).

$v_1 \; y_{12} \; v_2 \; y_{23} \; v_3 \; y_{34} \; v_4 \; y_{45} \; v_5$

$v_1 \; y_{13} \; v_3 \; y_{34} \; v_4 \; y_{46} \; v_6 \; y_{68} \; v_8$

(2) Same path with different labelling function, will also have distinct monomials.

$(W, l_1)$: $v_1 \; y_{12} \; v_2 \; y_{23} \; v_3 \; y_{34} \; v_4 \; y_{45} \; v_5$

$(W, l_2)$: $v_1 \; y_{12} \; v_2 \; y_{23} \; v_3 \; y_{34} \; v_4 \; y_{45} \; v_5$

**Conclude:** $\exists$ a $k$-path iff $p(\mathbf{x}, \mathbf{y}) \neq 0$.

Step 1: (done)

Reduce k-path to checking if $p(\mathbf{x},\mathbf{y})=0$?

Step 2:

How to check if $p(\mathbf{x},\mathbf{y}) = 0$?

Is $p(\mathbf{x}, \mathbf{y}) = 0$?

- degree of $p(\cdot) = K + (K-1) = 2K-1$

[The following is true for general polynomials]

If $p(\cdot)$ was a univariate polynomial?

$p(x) = x^2 + x$

<u>High school</u>: A non-zero univariate polynomial of degree d over the reals has at most d roots.

Multivariate polynomials can have arbitarily many roots,

Ex: $p(x_1, x_2) = x_1 x_2$

## <u>Schwartz - Zippel Lemma (Informal):</u>

Roots of multivariate polynomials are sparse, that is, when we randomly sample the values of the variables from a finite subset of the field, the probability of finding the root is small.

## Swartz - Zippel Lemma:

Let $p(x_1, \ldots, x_n)$ be a polynomial of degree $d$ over a field $F$, such that $p$ is not identically zero.

Let $S$ be a finite subset of $F$. Sample values $a_1, \ldots, a_n$ from $S$ uniformly at random.

Then,
$$\Pr\left(p(a_1, a_2, \ldots, a_n) = 0\right) \leq \frac{d}{|S|}.$$

**Proof:** Induction on $n$

$n=1$: $p(x_1)$ is a univariate polynomial of degree $d$.
Show that $p(x_1)$ has $\leq d$ roots. Then,
$\Pr(p(a_1) = 0) \leq d/|S|$.

$\underline{n>1}$: Rewrite

$$p(x_1, \dots, x_n) = \sum_{i=0}^{d} x_1^i \cdot p_i(x_2, \dots, x_n)$$

- Let $k \in \{0, 1, \dots, d\}$ be the largest index such that $p_k(x_2, \dots, x_n)$ is not identically zero. (Such a $k$ exists).

degree of $p_k(x_2, \dots, x_n) \leq d-k$

- For sampled values $a_2, \dots, a_n$ define a univariate polynomial,

$$q(x_1) = \sum_{i=0}^{d} x_1^i \cdot p_i(a_2, \dots, a_n)$$

- Want to compute $Pr(q(a_1)=0)$.

( $Pr(p(a_1, \dots, a_n)=0) = Pr(q(a_1)=0)$ )

Let $\mathcal{E}_1$ be the event that $q(a_1)=0$, and $\mathcal{E}_2$ be the event that $p_k(a_1, \dots, a_n) = 0$

$$
\begin{aligned}
\Pr(p(a_1, \ldots, a_n) = 0) &= \Pr(\mathcal{E}_1) \\
&= \Pr(\mathcal{E}_1 \cap \mathcal{E}_2) + \Pr(\mathcal{E}_1 \cap \overline{\mathcal{E}_2}) \\
&= \Pr(\mathcal{E}_2)\Pr(\mathcal{E}_1 | \mathcal{E}_2) + \Pr(\overline{\mathcal{E}_2})\Pr(\mathcal{E}_1 | \overline{\mathcal{E}_2}) \\
&\leq \Pr(\mathcal{E}_2) + \Pr(\mathcal{E}_1 | \overline{\mathcal{E}_2}) \\
&\leq \frac{d-k}{|S|} + \frac{k}{|S|} = \frac{d}{|S|}.
\end{aligned}
$$

- $\Pr(\mathcal{E}_2) = \Pr(p_k(a_2, \ldots, a_n) = 0) \leq \frac{d-k}{|S|}$

  from induction hypothesis.

- $\Pr(\mathcal{E}_1 | \overline{\mathcal{E}_2})$

  $\overline{\mathcal{E}_2} \Rightarrow p_k(a_2, \ldots, a_n) \neq 0$

If $\overline{\mathcal{E}_2}$ happens when $q(x_1)$ is not identically zero

$\because$ coefficient of $x_1^k$ is $p_k(a_2, \ldots, a_n) \neq 0$.

$\therefore$ $q(x_1)$ is a univariate polynomial that is not identically zero.

Also, degree of $q(x_1)$ is $k$ ($\because$ of choice of $k$).

$\therefore$ By induction hypothesis, $\Pr(\mathcal{E}_1 | \overline{\mathcal{E}_2}) \leq k / |S|$. ∎

## Back to our polynomial $p(x,y)$

degree of $p(\cdot) = k + k-1$.
$= 2k-1$

### Algorithm for k-path

- Evaluate $p(\cdot)$ over random values from the field $(F_{2^{\log 4k}}, +, \cdot)$.
- If evaluation makes $p(\cdot)=0 \Rightarrow$ return No (no k-path).
- Otherwise, return Yes.

### Correctness:

(1) If we return Yes $\Rightarrow p(\cdot)$ is not identically zero $\Rightarrow$ there is a k-path.

(2) If there is a k-path $\Rightarrow p(\cdot)$ is not identically zero.
degree of $p(\cdot) \leq 2k-1 \xRightarrow{\text{S-Z lemma}}$ with probability $\leq \frac{2k-1}{2^{\log 4k}} \leq \frac{1}{2}$ the evaluation makes $p(\cdot)$ zero.

$\geq 1/2$

**Theorem:** There is a one-sided error Monte Carlo algorithm with false negatives that solves k-path in time $O^*(2^k)$.

# How to evaluate our polynomial at given values?

$$p(\mathbf{x},\mathbf{y}) = \sum_{\substack{W=v_1\dots v_k \\ \text{is a walk}}} \; \sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is a bijection}}} \prod_{i=1}^{k} x_{v_i,\ell(i)} \prod_{i=1}^{k-1} y_{v_i,v_{i+1}}$$

Use dynamic programming.
(similar to the color coding DP)

For each $Z \subseteq [k]$ and $v \in V(G)$, compute

$$T[Z,v] = \sum_{\substack{\text{Walk} \\ W=v_1\dots v_{|Z|} \\ v_1 = v}} \; \sum_{\ell:[|Z|]\to Z} \prod_{i=1}^{|X|} x_{v_i,\ell(i)} \prod_{i=1}^{|X|-1} y_{v_i,v_{i+1}}$$

informally

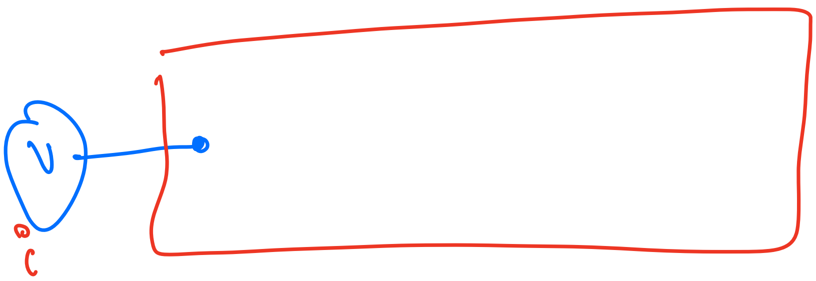
$$= \sum_{\substack{\text{labelled walks} \\ (w,\ell) \text{ of length } |Z| \\ \text{\& labels from color set } Z}} \mathrm{mon}(w,\ell)$$

Then $p(\mathbf{x},\mathbf{y}) = \sum_{v\in V(G)} T[[k],v]$.

# Computing $T[Z, v]$

$$T[Z, v] = \sum_{i \in Z} x_{v,i} \qquad \text{if } |Z| = 1$$

$$= \sum_{i \in Z} x_{v,i} \sum_{u \in N(v)} y_{v,u} \cdot T[u, Z \setminus \{i\}] \qquad \text{otherwise}$$




Each $T[Z, v]$ can be computed in polynomial time

$\Downarrow$

$p(\mathbf{x}, \mathbf{y})$ can be evaluated in $O^*(2^k)$ time

# Weighted inclusion-exclusion principle

Let $A_1, \ldots, A_n \subseteq \mathcal{U}$.
Let $w : \mathcal{U} \to \mathbb{R}$ and for any $X \subseteq \mathcal{U}$,
$w(X) = \sum_{x \in X} w(x)$.

Denote $\bigcap_{i \in \emptyset} (\mathcal{U} \setminus A_i) = \mathcal{U}$.

Then,

$$w\left(\bigcap_{i \in [n]} A_i\right) = \sum_{X \subseteq [n]} (-1)^{|X|} \cdot w\left(\bigcap_{i \in X} \overline{A}_i\right)$$

- Using weighted inclusion-exclusion principle, together with the previous ideas, one can show that

There is a one-sided error Monte Carlo algorithm with false negatives that solves $k$-path (also for directed graphs) in time $O^*(2^k)$ and polynomial space.