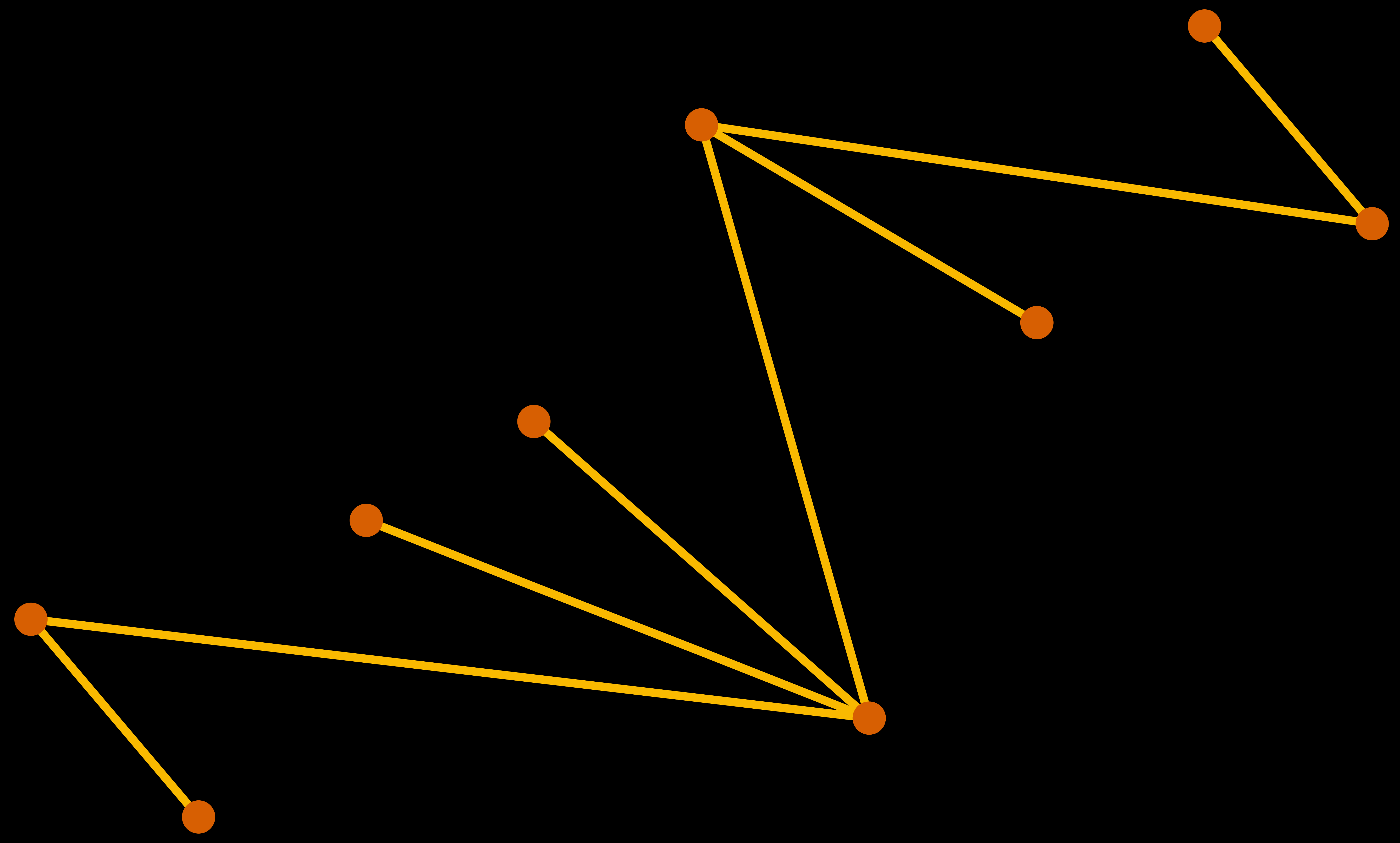


Inversions and Graphs

Sean Mandrick

Oral Examination - March 19th, 2024

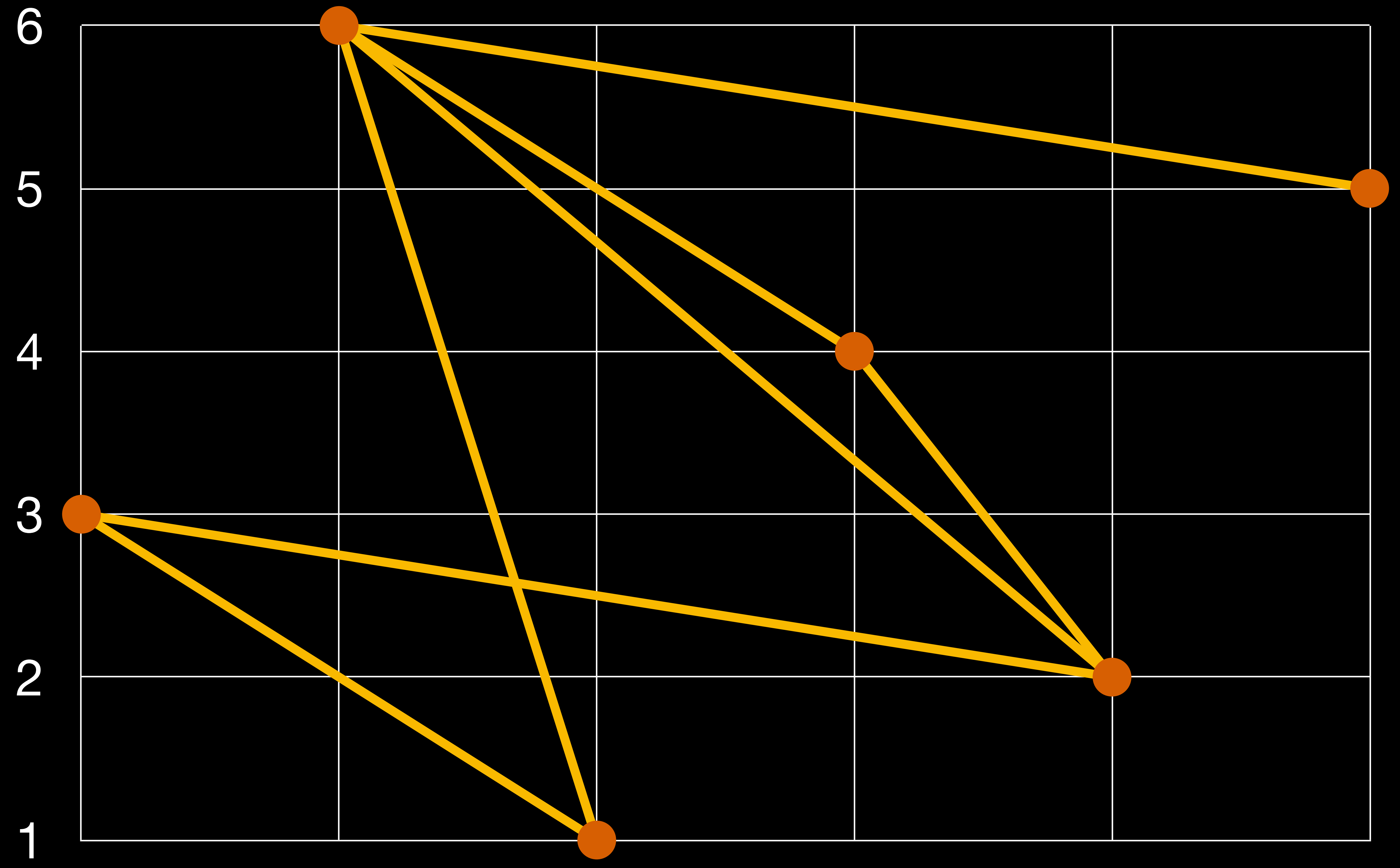
Preliminaries

- For a permutation $\pi \in S_n$, an ***inversion*** is a pair $(i, j),\ i, j \in [n]$, such that $i < j$ and $\pi(i) > \pi(j)$.

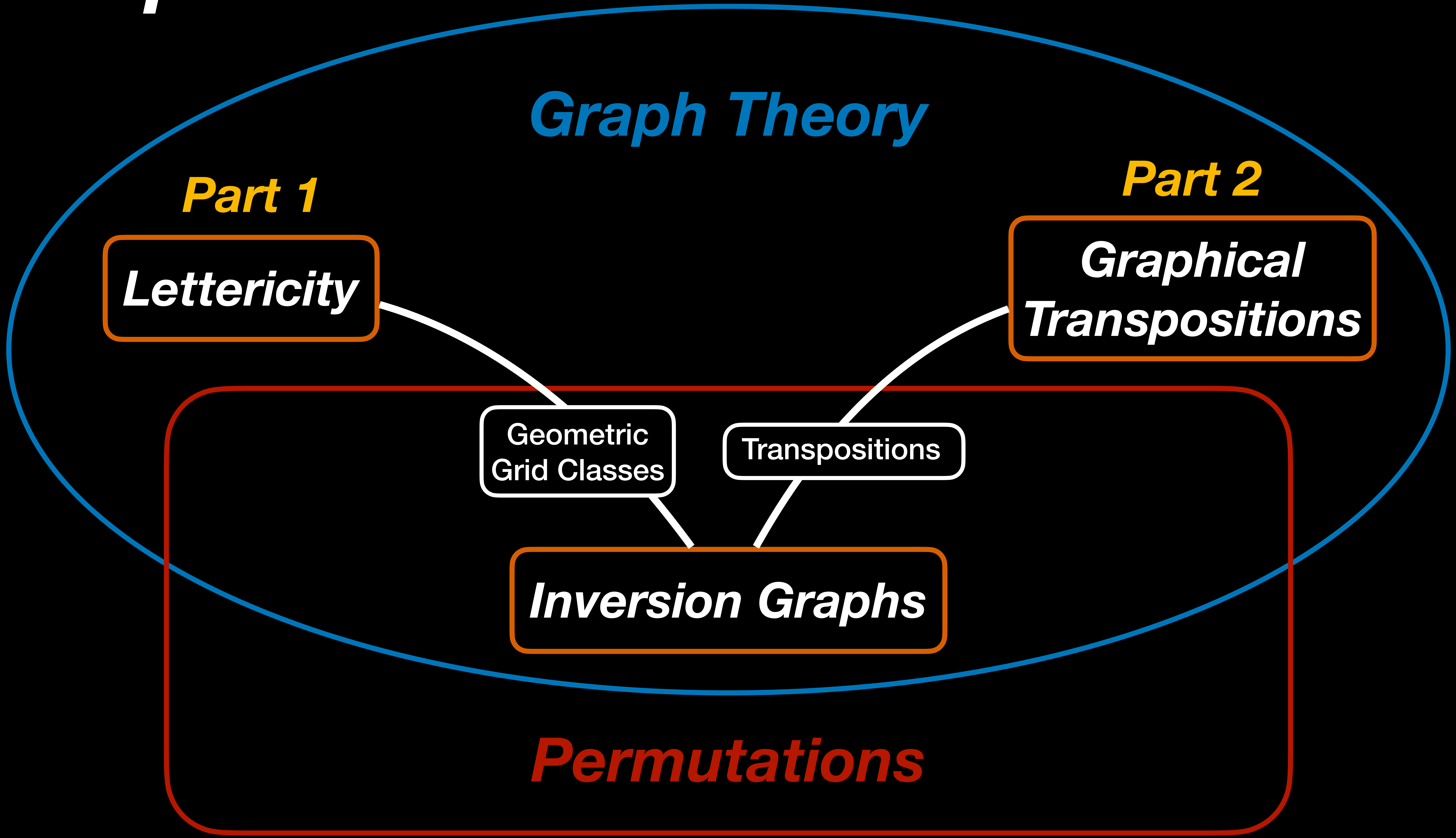
- Further, the ***inversion graph*** of the permutation $\pi \in S_n$, denoted G_π, is the graph with vertices $V(G_\pi) = [n]$ and edges given by the inversions of π, i.e. $E(G_\pi) = \{\pi(i)\pi(j) : (i, j) \text{ is an inversion of } \pi\}$.

An inversion graph example

- For a permutation $\pi \in S_n$, if we plot each vertex $\pi(i) \in [n]$ at $(i, \pi(i))$ in the plane, we obtain the inversion graph G_π by connecting each vertex to every vertex that is below and to its right. See G_{361425} below:



The map


Graph Theory
Part 1
Letticity
Part 2
Graphical Transpositions
Geometric Grid Classes
Transpositions
Inversion Graphs
Permutations

Part 1

Results on maximum lettericity

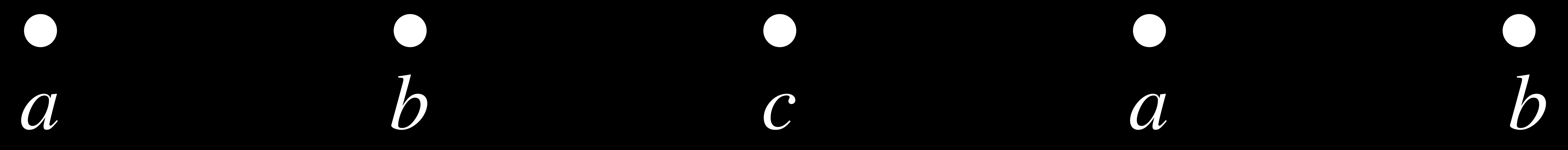
Letter graphs

- For a finite alphabet Σ, we consider a set of ordered pairs $D \subseteq \Sigma^2$ which we refer to as a ***decoder***.

- Then for a word $w = w_1 w_2 \ldots w_n$ with each $w_i \in \Sigma$, we define the ***letter graph*** of w to be the graph $\Gamma_D(w)$ with $V(\Gamma_D(w)) = \{1, 2, \ldots, n\}$ and for each pair $i < j$, we have $ij \in E(\Gamma_D(w))$ if and only if $(w_i, w_j) \in D$.

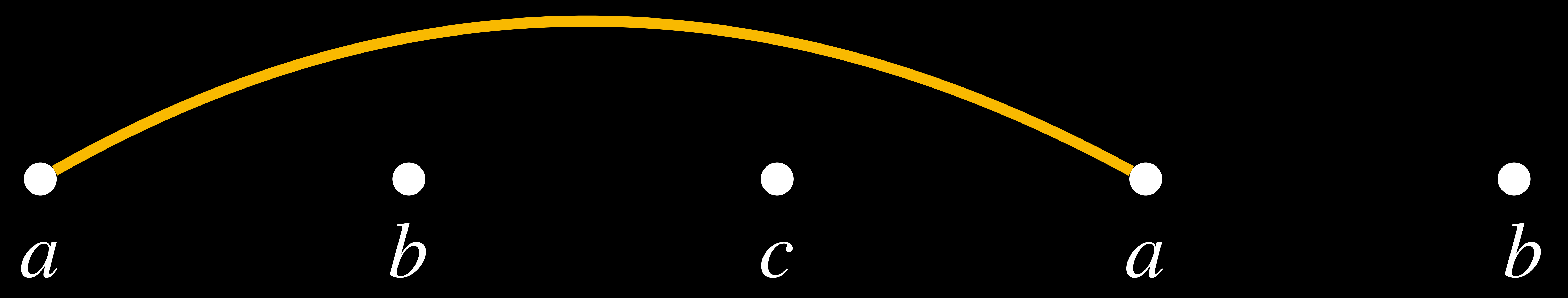
A visual example

- Let's consider the word $w = abcab$, and the decoder D with the tuples (a, a), (a, b) and (c, b). Then we can draw the graph $\Gamma_D(w)$ as follows:



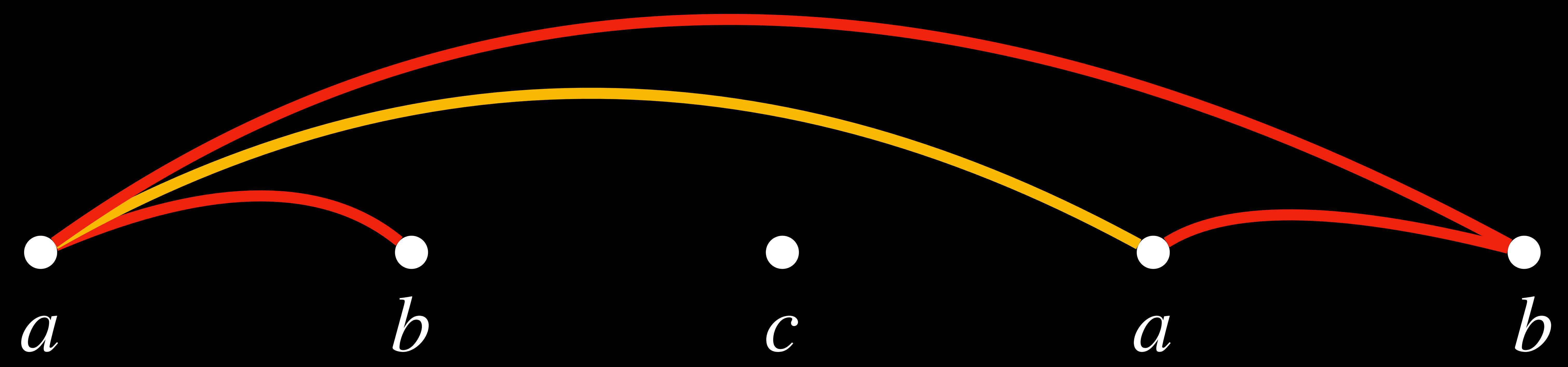
A visual example

- Let's consider the word $w = abcab$, and the decoder D with the tuples (a, a), (a, b) and (c, b). Then we can draw the graph $\Gamma_D(w)$ as follows:



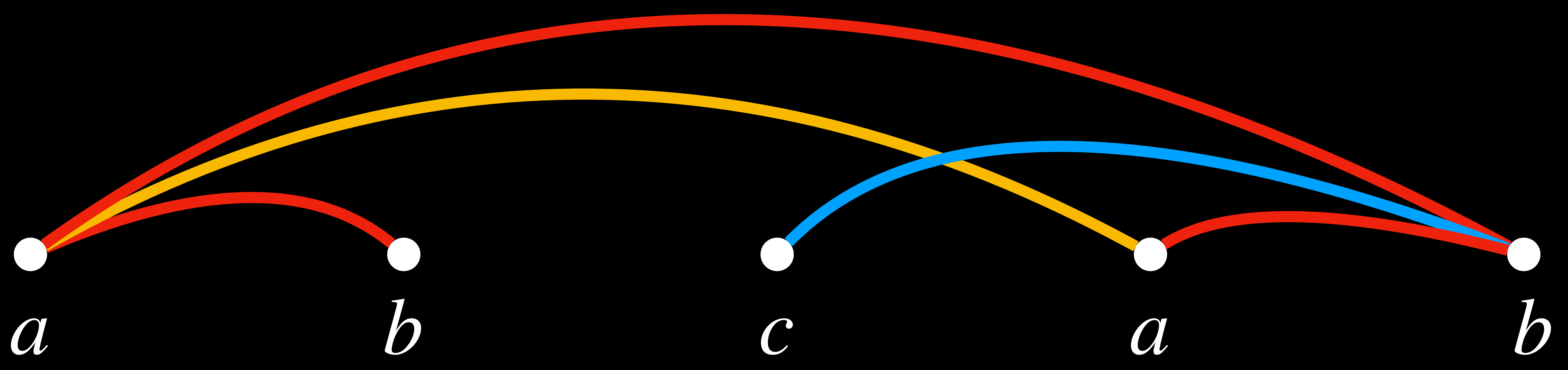
A visual example

- Let's consider the word $w = abcab$, and the decoder D with the tuples (a, a), (a, b) and (c, b). Then we can draw the graph $\Gamma_D(w)$ as follows:



A visual example

- Let's consider the word $w = abcab$, and the decoder D with the tuples (a, a), (a, b) and (c, b). Then we can draw the graph $\Gamma_D(w)$ as follows:



Letterity

- If $|\Sigma| = k$, then we say that $\Gamma_D(w)$ is a k-letter graph.

- Then, for any graph G, the least integer k such that G is isomorphic to a k-letter graph is called the ***lettericity*** of G, denoted $\ell(G)$.

- That is, the least size of an alphabet that admits the graph G for some word and decoder on that alphabet.

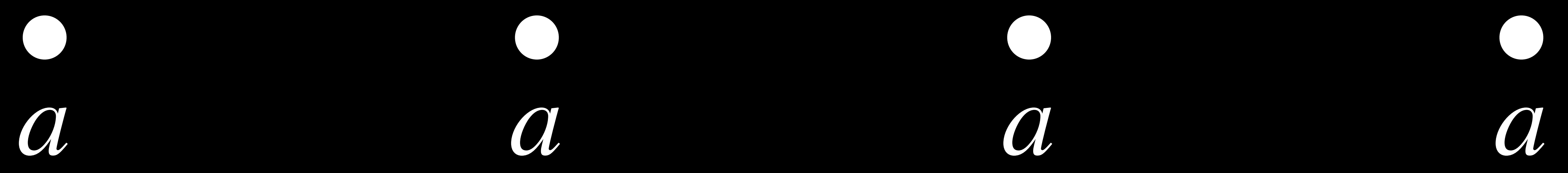
- This graph statistic, as well as the letter graph construction, were introduced by M. Petkovšek in the paper ***Letter graphs and well-quasi-order by induced subgraphs*** (2002).

Graphs with lettericity one

- If $\Sigma = \{a\}$, then there are only two possibilities:

1. $\Gamma_{\{(a,a)\}}(aa\ldots a) \;=\; K_n$

2. $\Gamma_{\varnothing}(aa\ldots a) \;=\; \overline{K_n}$



Threshold graphs have lettericity two

- A ***threshold graph*** is constructed by iteratively adding either dominating vertices (adjacent to all previously added vertices) or isolated vertices (adjacent to none of the previously added vertices).

- So with alphabet $\Sigma = \{i, d\}$ and decoder $D = \{(i, d), (d, d)\}$, we can think of the correspondences:

$$i \;\rightarrow\; \text{isolated vertices}$$
$$d \;\rightarrow\; \text{dominating vertices}$$

- That is, we draw $\Gamma_D(iddiid)$ by reading the word from left to right:

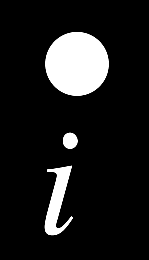
Threshold graphs have lettericity two

- A ***threshold graph*** is constructed by iteratively adding either dominating vertices (adjacent to all previously added vertices) or isolated vertices (adjacent to none of the previously added vertices).

- So with alphabet $\Sigma = \{i, d\}$ and decoder $D = \{(i, d), (d, d)\}$, we can think of the correspondences:

$$i \rightarrow \text{isolated vertices}$$
$$d \rightarrow \text{dominating vertices}$$

- That is, we draw $\Gamma_D(iddiid)$ by reading the word from left to right:



Threshold graphs have lettericity two

- A ***threshold graph*** is constructed by iteratively adding either dominating vertices (adjacent to all previously added vertices) or isolated vertices (adjacent to none of the previously added vertices).

- So with alphabet $\Sigma = \{i, d\}$ and decoder $D = \{(i, d), (d, d)\}$, we can think of the correspondences:

$$i \rightarrow \text{isolated vertices}$$
$$d \rightarrow \text{dominating vertices}$$

- That is, we draw $\Gamma_D(iddiid)$ by reading the word from left to right:

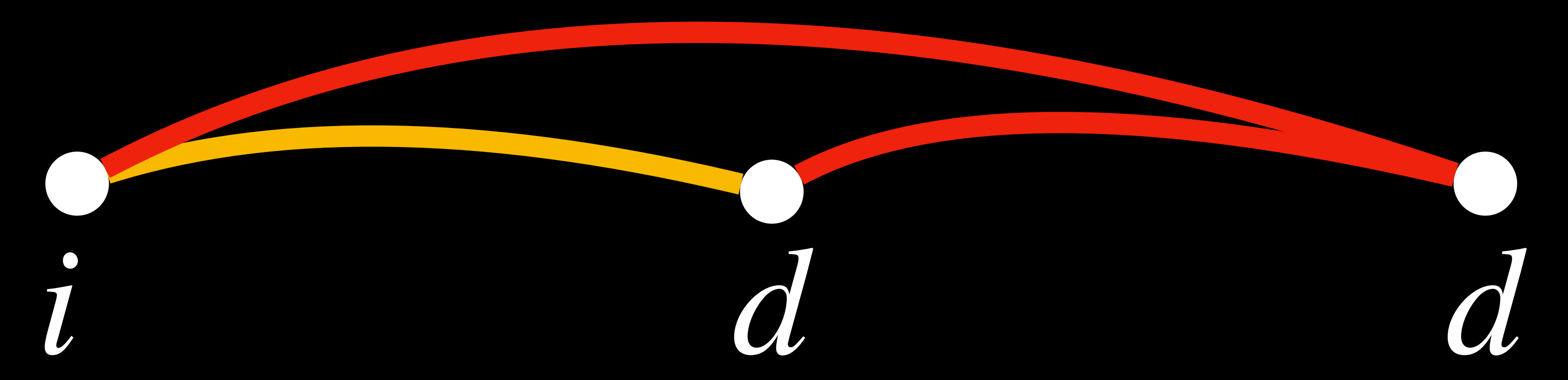
Threshold graphs have lettericity two

- A ***threshold graph*** is constructed by iteratively adding either dominating vertices (adjacent to all previously added vertices) or isolated vertices (adjacent to none of the previously added vertices).

- So with alphabet $\Sigma = \{i, d\}$ and decoder $D = \{(i, d), (d, d)\}$, we can think of the correspondences:

$$i \rightarrow \text{isolated vertices}$$
$$d \rightarrow \text{dominating vertices}$$

- That is, we draw $\Gamma_D(iddiid)$ by reading the word from left to right:



Threshold graphs have lettericity two

- A ***threshold graph*** is constructed by iteratively adding either dominating vertices (adjacent to all previously added vertices) or isolated vertices (adjacent to none of the previously added vertices).

- So with alphabet $\Sigma = \{i, d\}$ and decoder $D = \{(i,d),(d,d)\}$, we can think of the correspondences:

$$i \rightarrow \text{isolated vertices}$$
$$d \rightarrow \text{dominating vertices}$$

- That is, we draw $\Gamma_D(iddiid)$ by reading the word from left to right:

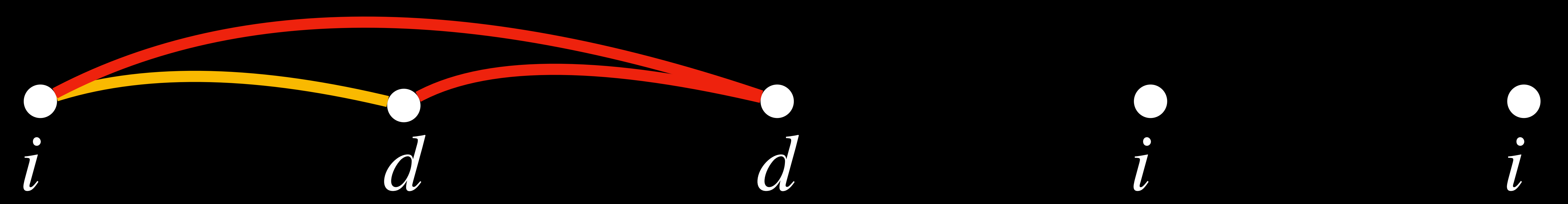
Threshold graphs have lettericity two

- A ***threshold graph*** is constructed by iteratively adding either dominating vertices (adjacent to all previously added vertices) or isolated vertices (adjacent to none of the previously added vertices).

- So with alphabet $\Sigma = \{i, d\}$ and decoder $D = \{(i, d), (d, d)\}$, we can think of the correspondences:

$$i \rightarrow \text{isolated vertices}$$
$$d \rightarrow \text{dominating vertices}$$

- That is, we draw $\Gamma_D(iddiid)$ by reading the word from left to right:



Threshold graphs have lettericity two

- A ***threshold graph*** is constructed by iteratively adding either dominating vertices (adjacent to all previously added vertices) or isolated vertices (adjacent to none of the previously added vertices).

- So with alphabet $\Sigma = \{i, d\}$ and decoder $D = \{(i, d), (d, d)\}$, we can think of the correspondences:

$$i \rightarrow \text{isolated vertices}$$
$$d \rightarrow \text{dominating vertices}$$

- That is, we draw $\Gamma_D(iddiid)$ by reading the word from left to right:

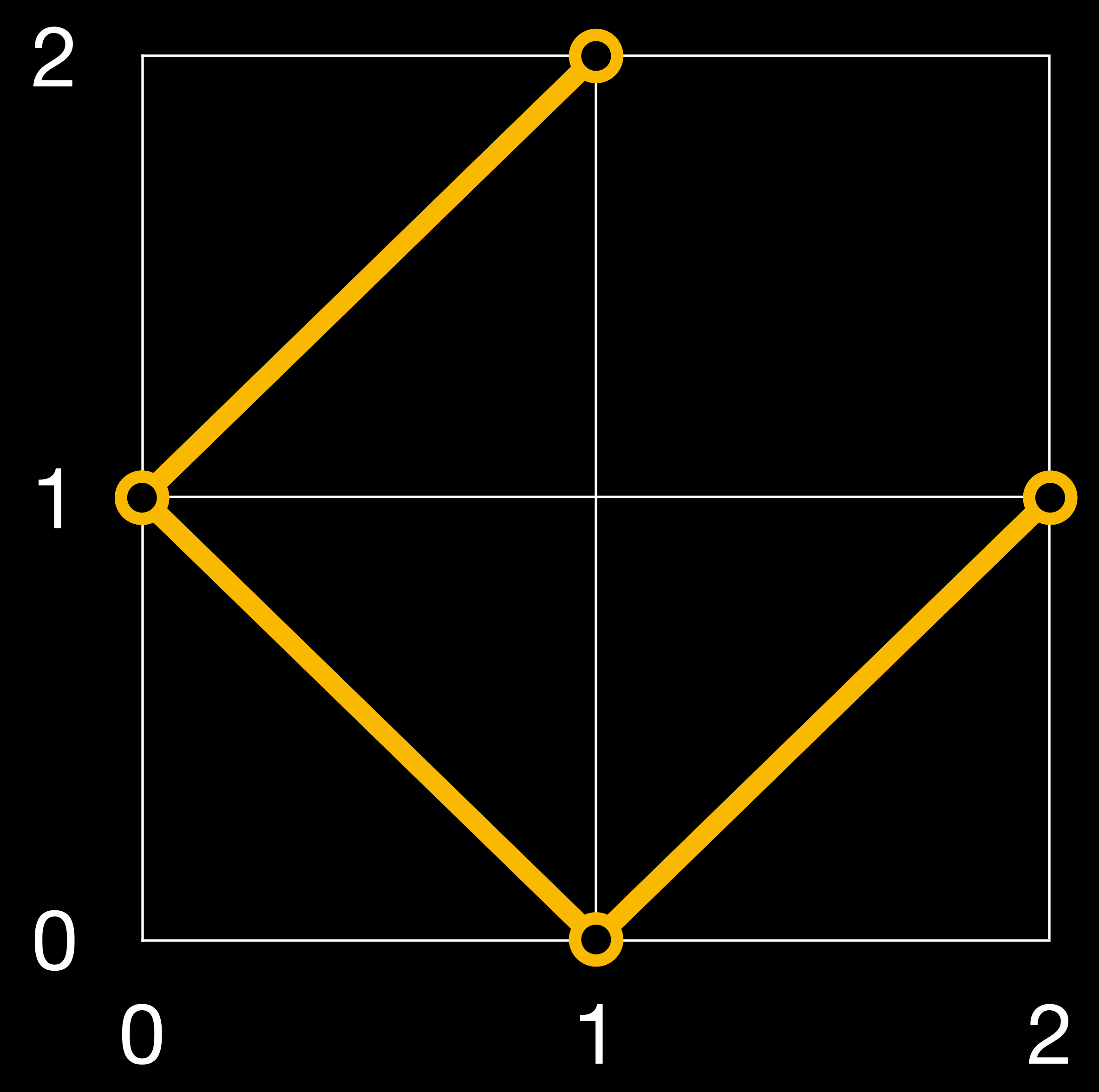
Geometric grid class preliminaries

- The ***standard figure*** of a $0/\pm 1$ matrix $M = (m_{i,j})$ is the point set in $\mathbb{R}^2$ consisting of

  - the increasing open line segment from $(i-1, j-1)$ to (i, j) if $m_{i,j} = 1$ or

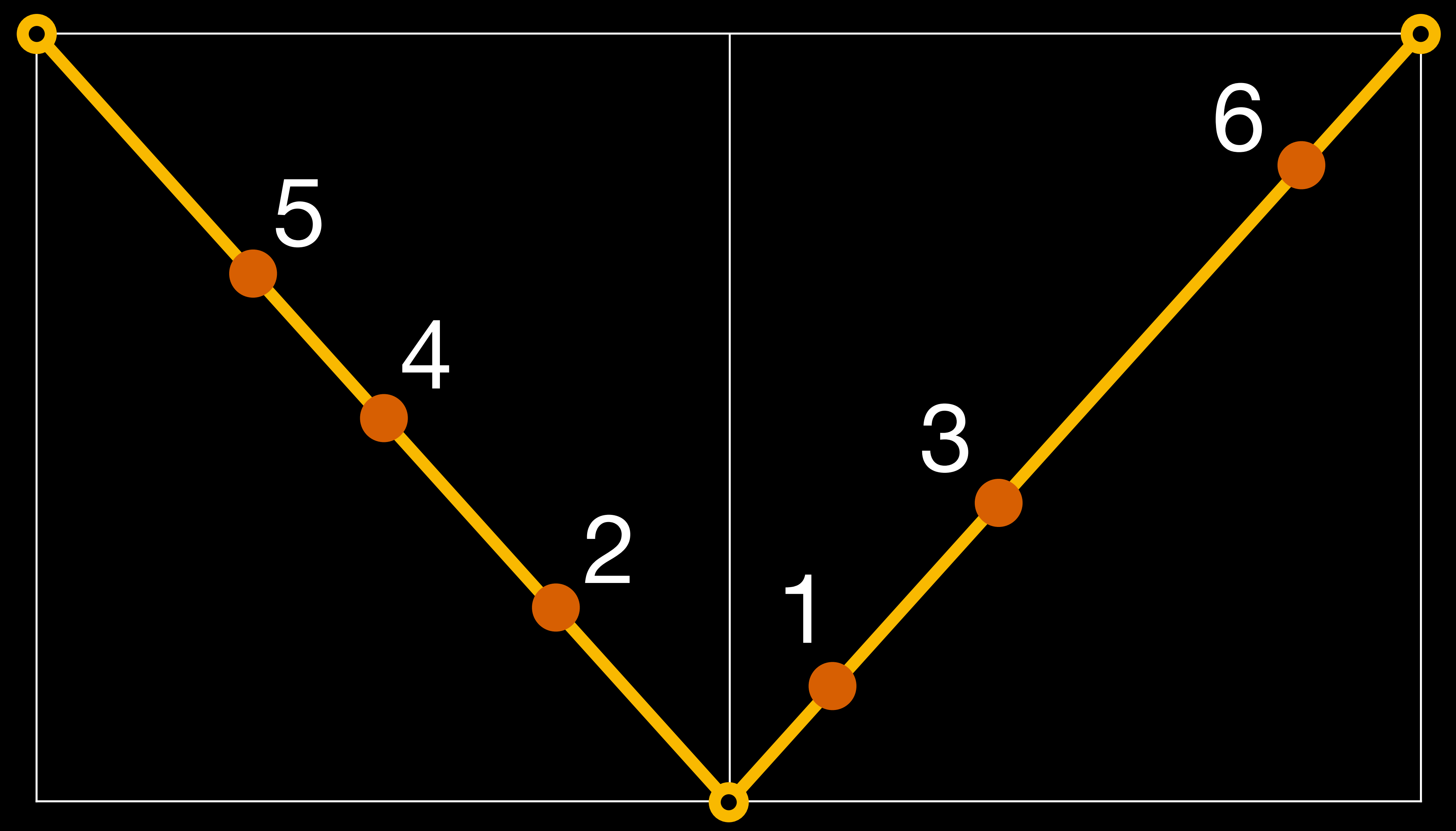
  - the decreasing open line segment from $(i-1, j)$ to $(i, j-1)$ if $m_{i,j} = -1$.

- For example, $M = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$ has standard figure:



Geometric grid classes

- The ***geometric grid class*** of M, denoted Geom(M), is the class of all permutations that can be 'properly' drawn on the standard figure of M.
- By 'properly', we mean that for each pair of entries of the permutation, they are in the same relative positions as they are in the inversion graph.
- For example, we see that $542136 \in$ Geom $\begin{pmatrix} -1 & 1 \end{pmatrix}$:



Geometric grid classes and lettericity

- In the paper ***Letter Graphs and Geometric Grid Classes of Permutations*** (2022), Alecu, Ferguson, Kanté, Lozin, Vatter and Zamaraev showed that

> *"… the concepts of lettericity and geometric griddability capture the same structural data of their respective combinatorial objects…"*

- More specifically, they proved the following:

> ***Theorem:*** The permutation class $\mathscr{C}$ is geometrically griddable if and only if the corresponding graph class $G_{\mathscr{C}}$ has bounded lettericity.

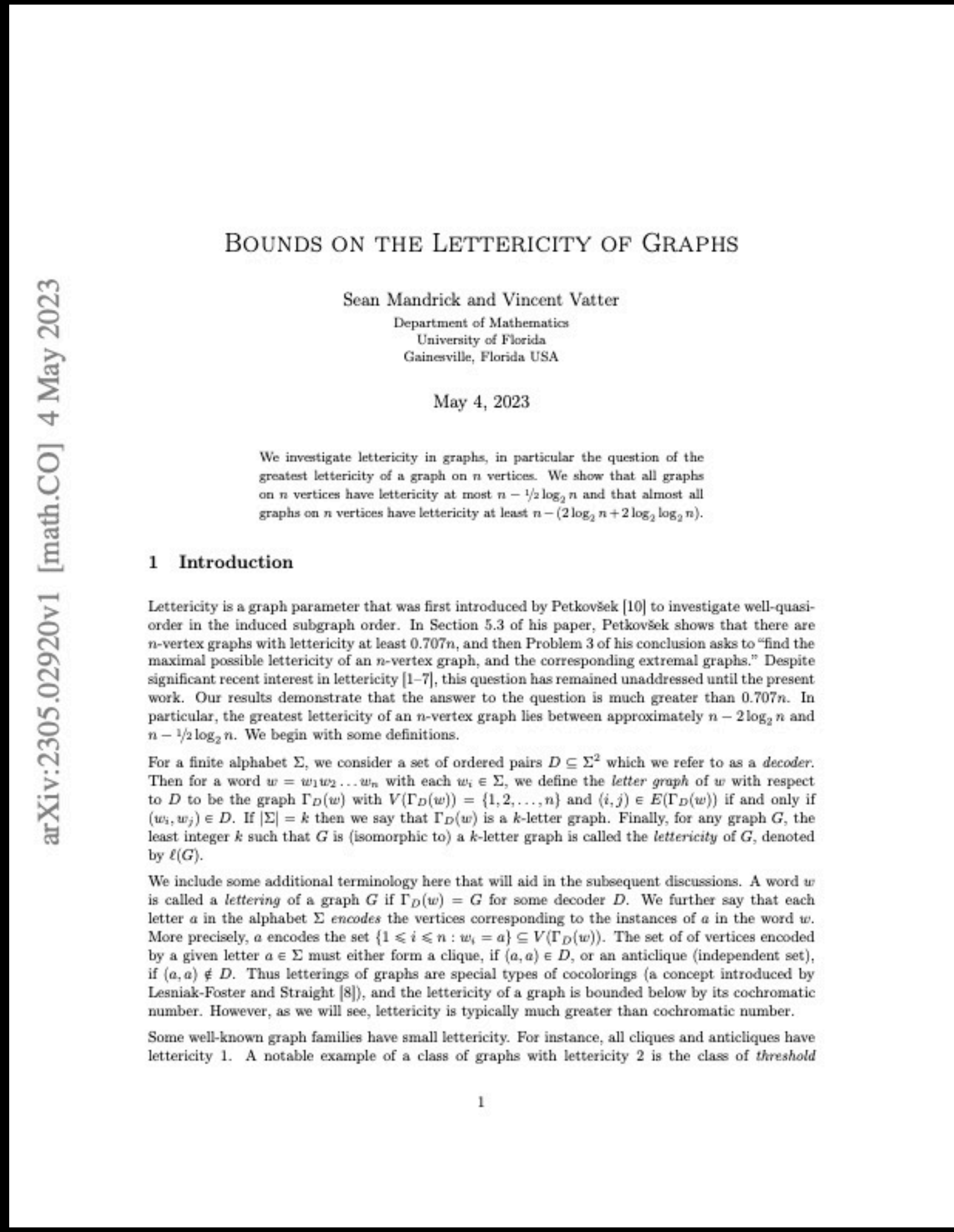
- We will return to these ideas later…

The problem

- In Petkovšek's paper introducing lettericity, he posed the following:

> ***Problem 3:*** Find the maximum possible lettericity of an n-vertex graph, and the corresponding extremal graphs.

- In the paper ***Bounds on the Lettericity of Graphs*** (2023) with V. Vatter, we significantly improve the known results pertaining to the maximum possible lettericity of an n-vertex graph.



arXiv:2305.02920v1 [math.CO] 4 May 2023

BOUNDS ON THE LETTERICITY OF GRAPHS

Sean Mandrick and Vincent Vatter
Department of Mathematics
University of Florida
Gainesville, Florida USA

May 4, 2023

We investigate lettericity in graphs, in particular the question of the greatest lettericity of a graph on n vertices. We show that all graphs on n vertices have lettericity at most $n - 1/2 \log_2 n$ and that almost all graphs on n vertices have lettericity at least $n - (2\log_2 n + 2\log_2 \log_2 n)$.

1 Introduction

Lettericity is a graph parameter that was first introduced by Petkovšek [10] to investigate well-quasi-order in the induced subgraph order. In Section 5.3 of his paper, Petkovšek shows that there are n-vertex graphs with lettericity at least $0.707n$, and then Problem 3 of his conclusion asks to "find the maximal possible lettericity of an n-vertex graph, and the corresponding extremal graphs." Despite significant recent interest in lettericity [1–7], this question has remained unaddressed until the present work. Our results demonstrate that the answer to the question is much greater than $0.707n$. In particular, the greatest lettericity of an n-vertex graph lies between approximately $n - 2\log_2 n$ and $n - 1/2 \log_2 n$. We begin with some definitions.

For a finite alphabet Σ, we consider a set of ordered pairs $D \subseteq \Sigma^2$ which we refer to as a *decoder*. Then for a word $w = w_1 w_2 \dots w_n$ with each $w_i \in \Sigma$, we define the *letter graph* of w with respect to D to be the graph $\Gamma_D(w)$ with $V(\Gamma_D(w)) = \{1, 2, \dots, n\}$ and $(i, j) \in E(\Gamma_D(w))$ if and only if $(w_i, w_j) \in D$. If $|\Sigma| = k$ then we say that $\Gamma_D(w)$ is a k-letter graph. Finally, for any graph G, the least integer k such that G is (isomorphic to) a k-letter graph is called the *lettericity* of G, denoted by $\ell(G)$.

We include some additional terminology here that will aid in the subsequent discussions. A word w is called a *lettering* of a graph G if $\Gamma_D(w) = G$ for some decoder D. We further say that each letter a in the alphabet Σ *encodes* the vertices corresponding to the instances of a in the word w. More precisely, a encodes the set $\{1 \leqslant i \leqslant n : w_i = a\} \subseteq V(\Gamma_D(w))$. The set of of vertices encoded by a given letter $a \in \Sigma$ must either form a clique, if $(a, a) \in D$, or an anticlique (independent set), if $(a, a) \notin D$. Thus letterings of graphs are special types of cocolorings (a concept introduced by Lesniak-Foster and Straight [8]), and the lettericity of a graph is bounded below by its cochromatic number. However, as we will see, lettericity is typically much greater than cochromatic number.

Some well-known graph families have small lettericity. For instance, all cliques and anticliques have lettericity 1. A notable example of a class of graphs with lettericity 2 is the class of *threshold*

1

Previously known results

- The previously known bounds on maximum lettericity of an n-vertex graph are from Petkovšek's original paper:

Upper bound: For all graphs G on n vertices, we have $\ell(G) \leq n - 1$.

Lower bound: For each $\alpha < 1/\sqrt{2}$, there is an N such that for all $n > N$, there are n-vertex graphs G with $\ell(G) \geq \alpha n \approx 0.7071n$.

Proof of previous upper bound

- ***Upper bound:*** For all graphs G on n vertices, we have $\ell(G) \leq n - 1$.

Proof: It is clear that we can encode any n-vertex G with the word $a_1 a_2 \ldots a_n$ by adding the appropriate elements to the decoder D.

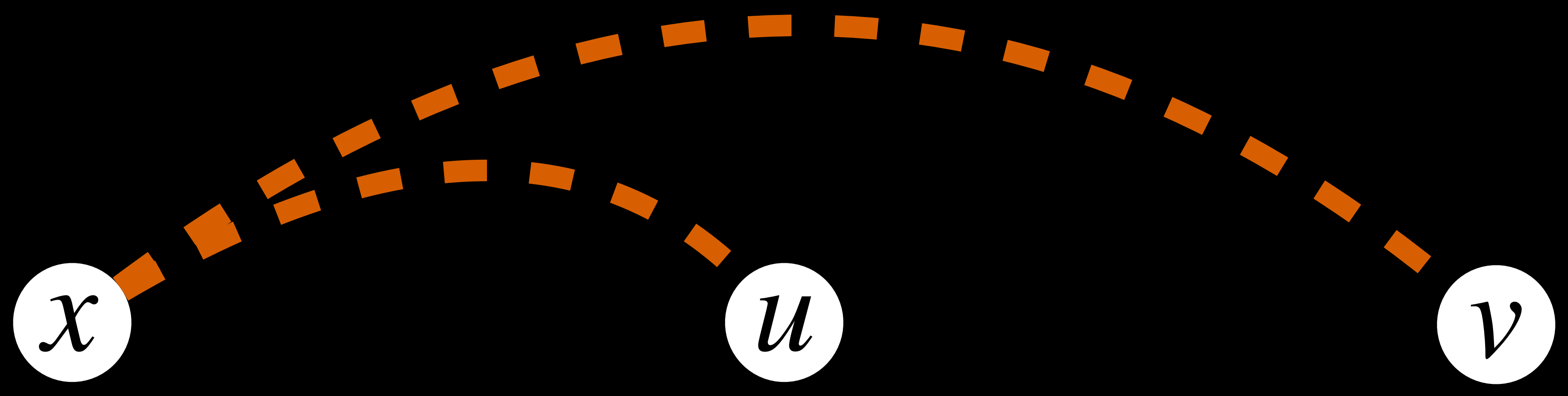
We can then swap all instances of a_n with a_1 in this word and decoder, and the new word and decoder will still encode the graph G, (with $n - 1$ letters).

Word	**Decoder Elements**
$a_1 a_2 \ldots a_{n-1} a_n$	(a_k, a_n)
↓	↓
$a_1 a_2 \ldots a_{n-1} a_1$	(a_k, a_1) ∎

The key observation

- We will expand on the main idea of this proof to obtain both the improved upper and lower bounds.

- That is, suppose we encode two vertices, u and v, with the same letter a. Then we can encode other vertices to the left or the right of both of the a's only if they agree on u and v:

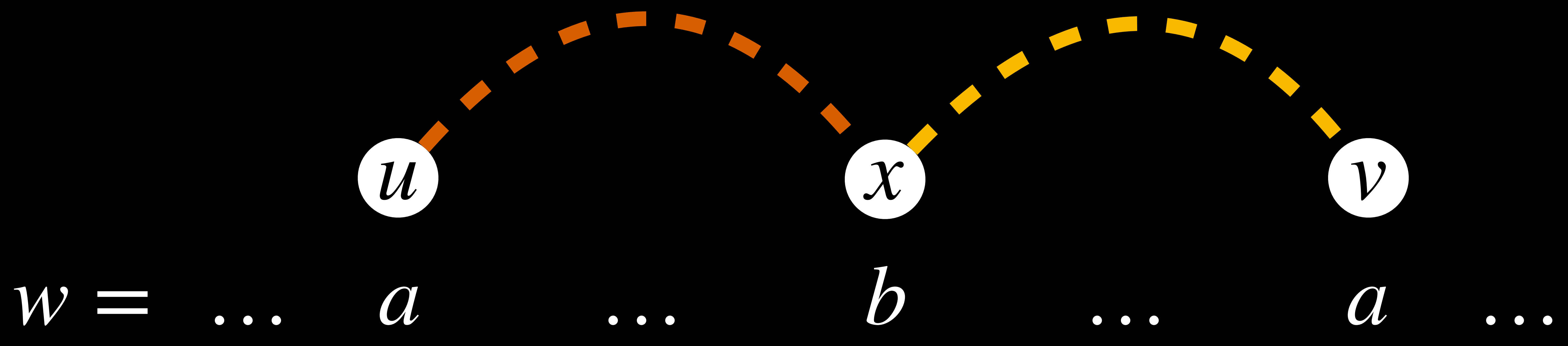


$$w = \dots \quad b \quad \dots \quad a \quad \dots \quad a \quad \dots$$

Both dashed edges are decided by presence of (b, a) in the decoder.

The key observation (2)

- But we can always encode other vertices between the two instances of a:



The left and right dashed edges are decided by (a, b) and (b, a), respectively.

- This is exactly what we did in the last proof… we saw that we can encode every graph with the word

$$a_1\, a_2\, a_3\, \ldots\, a_{n-1}\, a_1.$$

Construction for improved upper bound

- We will use the following construction to take advantage of this observation and obtain the improved upper bound.

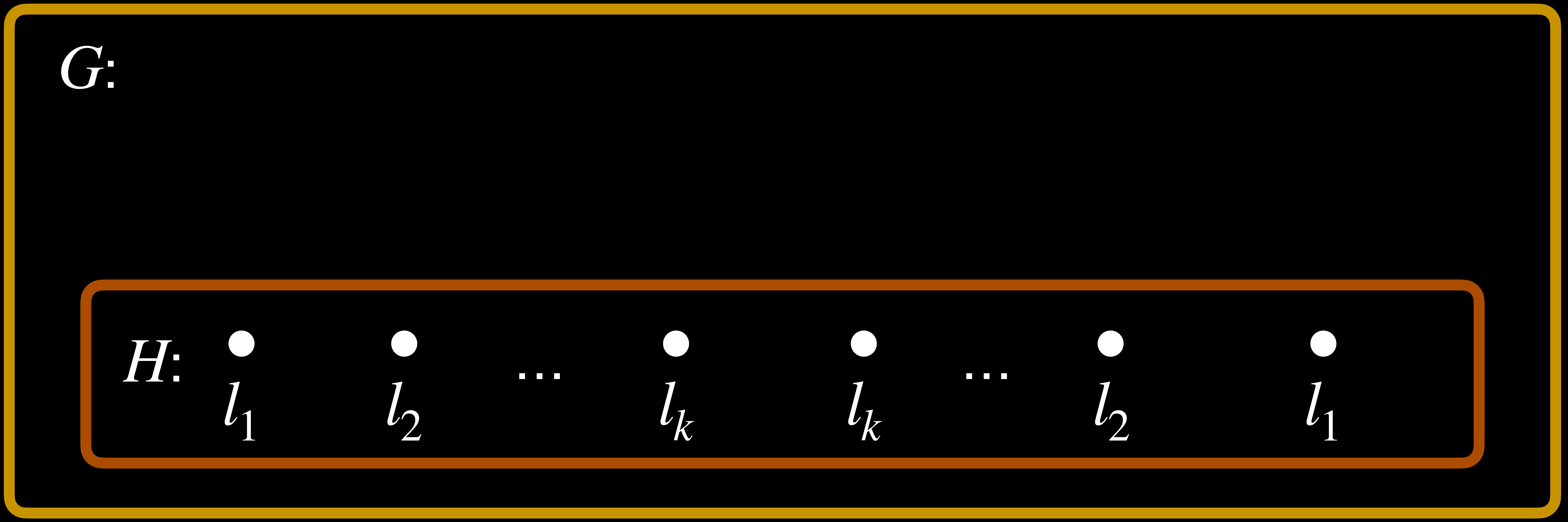
- ***Proposition:*** For every k and each graph G on $n \geq 2(k-1) + 2^{2(k-1)} + 1$ vertices, G has an induced subgraph with $2k$ vertices that is a k-letter graph on the word

$$w = l_1\, l_2 \ldots l_k\, l_k \ldots l_2\, l_1.$$

Proof of proposition (1)

- ***Proposition:*** For every k and each graph G on $n \geq 2(k-1) + 2^{2(k-1)} + 1$ vertices, G has an induced subgraph with $2k$ vertices that is a k-letter graph on the word $w = l_1\, l_2 \ldots l_k\, l_k \ldots l_2\, l_1$.

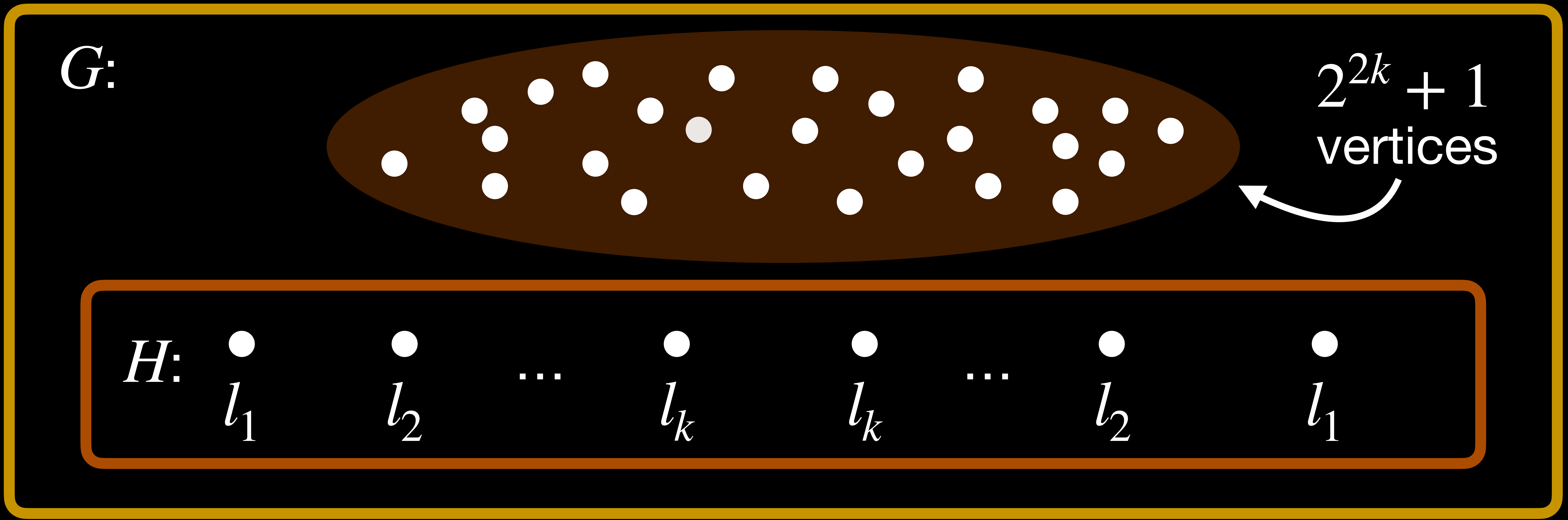
Proof: Induction on k. Base case $k = 1$ is just any two vertices. So we suppose G has at least $2k + 2^{2k} + 1$ vertices, and given the result holds for k we have:



Proof of proposition (1)

- ***Proposition:*** For every k and each graph G on $n \geq 2(k-1) + 2^{2(k-1)} + 1$ vertices, G has an induced subgraph with $2k$ vertices that is a k-letter graph on the word $w = l_1\, l_2 \ldots l_k\, l_k \ldots l_2\, l_1$.

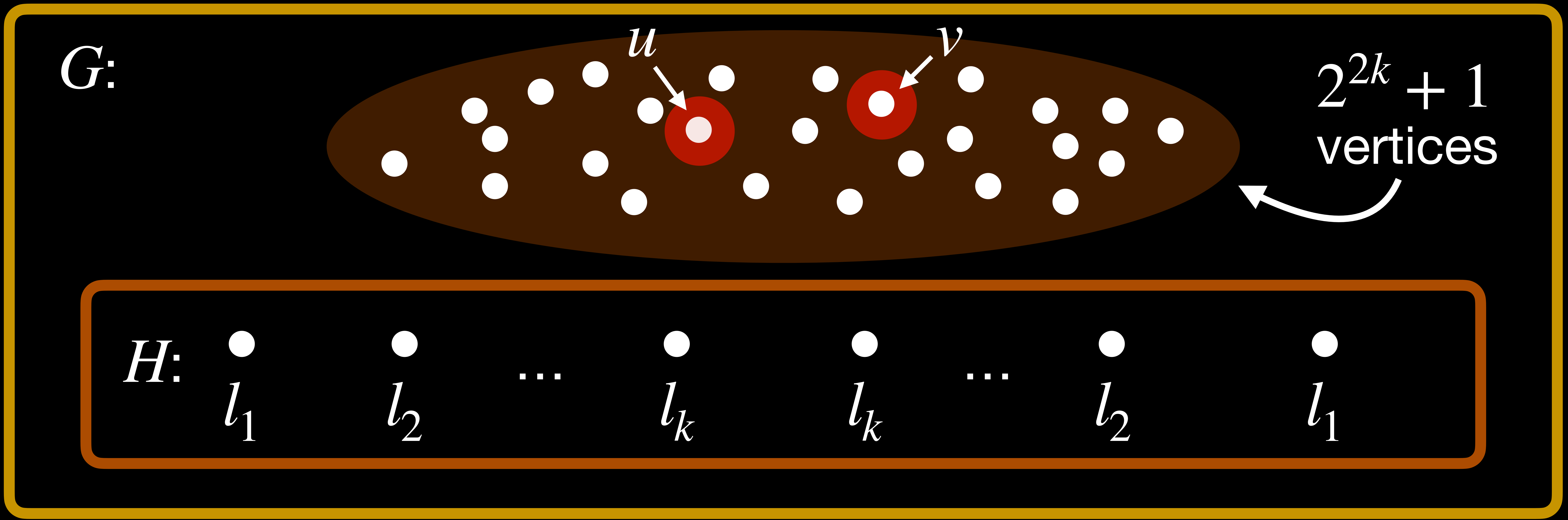
Proof: Induction on k. Base case $k = 1$ is just any two vertices. So we suppose G has at least $2k + 2^{2k} + 1$ vertices, and given the result holds for k we have:



Proof of proposition (1)

- **Proposition:** For every k and each graph G on $n \geq 2(k-1) + 2^{2(k-1)} + 1$ vertices, G has an induced subgraph with $2k$ vertices that is a k-letter graph on the word $w = l_1\, l_2 \ldots l_k\, l_k \ldots l_2\, l_1$.

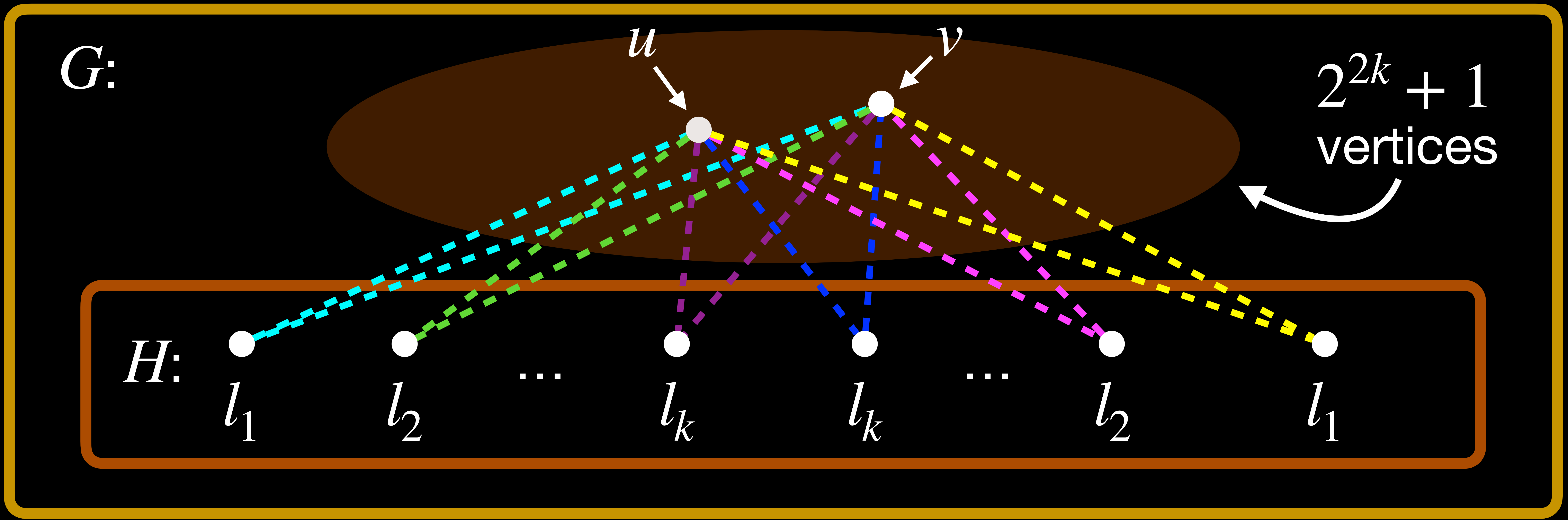
Proof: Induction on k. Base case $k = 1$ is just any two vertices. So we suppose G has at least $2k + 2^{2k} + 1$ vertices, and given the result holds for k we have:



Proof of proposition (1)

- ***Proposition:*** For every k and each graph G on $n \geq 2(k-1) + 2^{2(k-1)} + 1$ vertices, G has an induced subgraph with $2k$ vertices that is a k-letter graph on the word $w = l_1\, l_2 \ldots l_k\, l_k \ldots l_2\, l_1$.

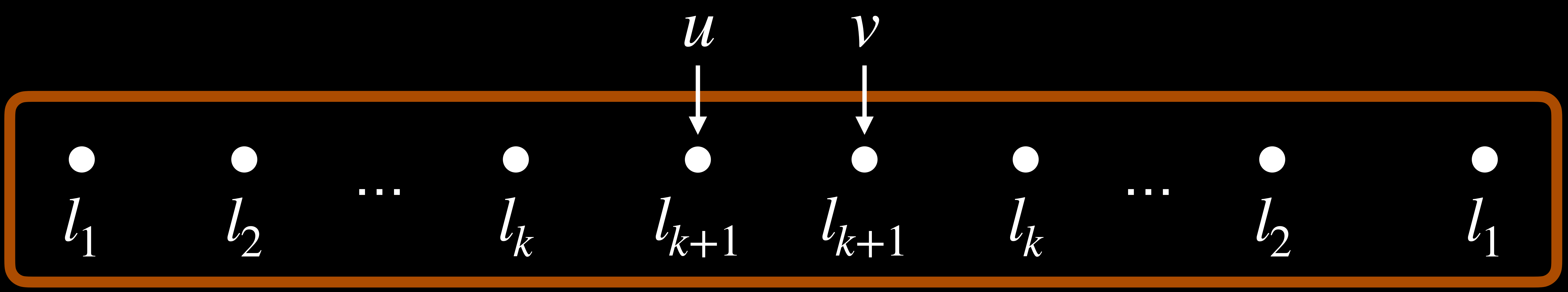
Proof: Induction on k. Base case $k = 1$ is just any two vertices. So we suppose G has at least $2k + 2^{2k} + 1$ vertices, and given the result holds for k we have:



Proof of proposition (2)

- ***Proposition:*** For every k and each graph G on $n \geq 2(k-1) + 2^{2(k-1)} + 1$ vertices, G has an induced subgraph with $2k$ vertices that is a k-letter graph on the word $w = l_1\, l_2 \ldots l_k\, l_k \ldots l_2\, l_1$.

Proof: (Continued) We can then safely extend H and the word w using these vertices u and v to obtain the desired construction:



∎

The improved upper bound

- **Theorem:** For every k and each graph G on $n \geq 2(k-1) + 2^{2(k-1)} + 1$ vertices, we have

$$\ell(G) \leq n - k.$$

Proof: We can encode the rest of the vertices in the middle of this construction, each with its own letter:

From the proposition

$$\boxed{l_1\, l_2 \ldots l_k}\, a_1\, a_2\, \ldots\, a_{n-2k}\, \boxed{l_k \ldots l_2\, l_1} \quad \blacksquare$$

- Essentially, we can 'save' about $(1/2)\log_2 n$ letters when encoding any graph G on n vertices.

Previous lower bound

- Recall, the previously known lower bound on maximum lettericity of a graph says that there exists n-vertex graphs with lettericity at least $0.707n$ for large enough n.

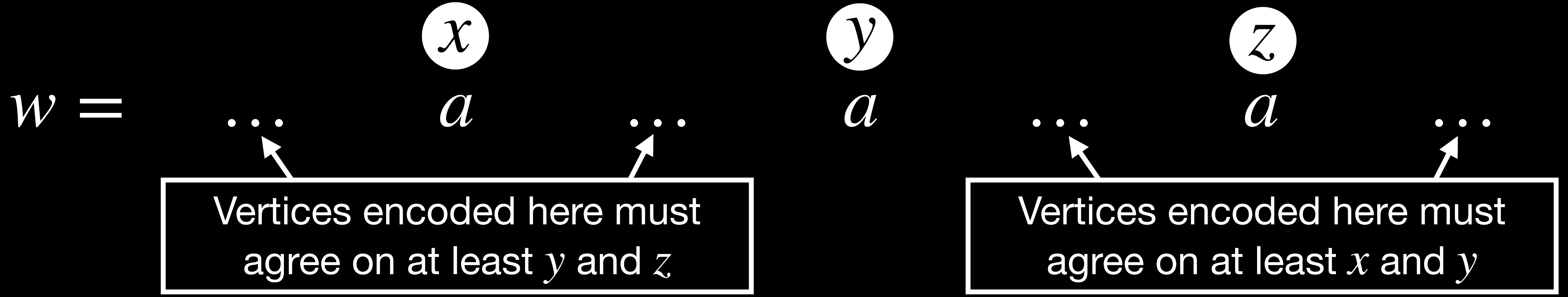
  Proof: Some not so enlightening inequalities…

- It turns out we can do much better than this result, as we are able to show that essentially all graphs are actually somewhat close to the new upper bound.

First fact for lower bound

- ***Fact 1:*** For almost all graphs G, no three vertices can be encoded by the same letter in a lettering of G.

Proof: Let $G = G(n, 1/2)$. Suppose we wish to encode three vertices x, y and z with the same letter in a lettering of G:



Letting $A_{(x,y,z)}$ denote the event that this is possible, we have $\mathbb{P}[A_{(x,y,z)}] \leq (3/4)^{n-3}$.

Then the probability that we can encode any three vertices with the same letter is

$$\mathbb{P}\left[\bigcup A_{(x,y,z)}\right] \leq \sum_{(x,y,z)} \mathbb{P}[A_{(x,y,z)}] \leq (n)_3 \cdot (3/4)^{n-3} \to 0 \text{ as } n \to \infty. \blacksquare$$

Second fact for lower bound

- We can now assume that at most two vertices can be encoded with one letter.

- Given this, if two letters, say a and b, each appear twice in a word, we define the following patterns for their possible relative positions:

  - ***Crossing:*** … a … b … a … b …
  - ***Nested:*** … a … b … b … a …
  - ***Separated:*** … a … a … b … b …

- ***Fact 2:*** For almost all graphs G, if two letters appear twice in a lettering of G, they must appear in a crossing or nested pattern.

Second fact for lower bound (2)

- ***Fact 2:*** For almost all graphs G, if two letters appear twice in a lettering of G, they must appear in a crossing or nested pattern.

Proof: We make almost the exact same argument. That is, suppose we try to encode $G = G(n,1/2)$ in a word with a separated pattern:

$$w = \ \dots a \dots a \dots b \dots b \dots$$

Then to encode a vertex somewhere in this word, it must either agree on the vertices encoded by a or the vertices encoded by b.

$$\vdots$$

The probability that we can encode G with a word containing a separated pattern goes to 0 as $n \to \infty$. ∎

How else can we save letters?

- In summary, for almost all graphs G on n vertices, if G has a lettering w with $n-k$ letters, then it will have k letters that appear twice and $n-2k$ letters that appear once.
- Furthermore, there exists a permutation $\pi \in S_k$ such that

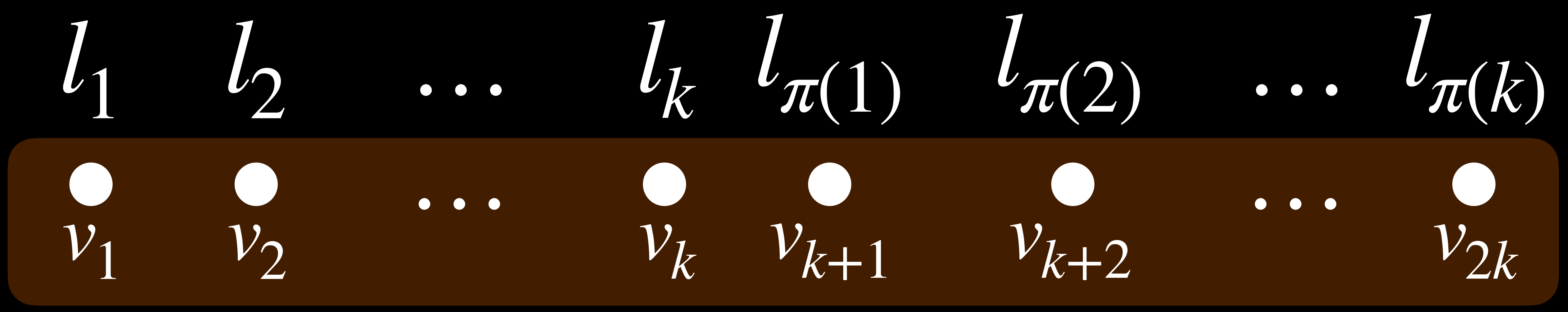
$$\omega = l_1\, l_2 \ldots l_k\, l_{\pi(1)}\, l_{\pi(2)} \cdots\, l_{\pi(k)}$$

  is the subword of w containing all of the letters that appear twice, (i.e. this accounts for all the ways for each pair of letters to be crossing or nested).

- It remains to analyze how small we can make this k (as a function of n) so that the probability that G contains an induced subgraph that can be encoded by a word ω goes to 0.

How else can we save letters? *(2)*

- So for some vertices $(v_i)_{i=1}^{2k} = (v_1, v_2, \ldots, v_{2k})$ of $G(n, 1/2)$ and a permutation $\pi \in S_k$, we calculate the probability that these vertices can be encoded in this order by the word



- Letting $C_{(v_i),\pi}$ denote this event, it turns out that $\mathbb{P}[C_{(v_i),\pi}] = (1/4)^{\binom{k}{2}}$.

- We again appeal to the union bound, this time to get an upper bound on the probability that any such construction is possible:

$$\mathbb{P}\left[\bigcup C_{(v_i),\pi}\right] \leq \sum_{(v_i),\pi} \mathbb{P}[C_{(v_i),\pi}] = (n)_{2k} \cdot k! \cdot 2^{-k(k-1)}.$$

The improved lower bound

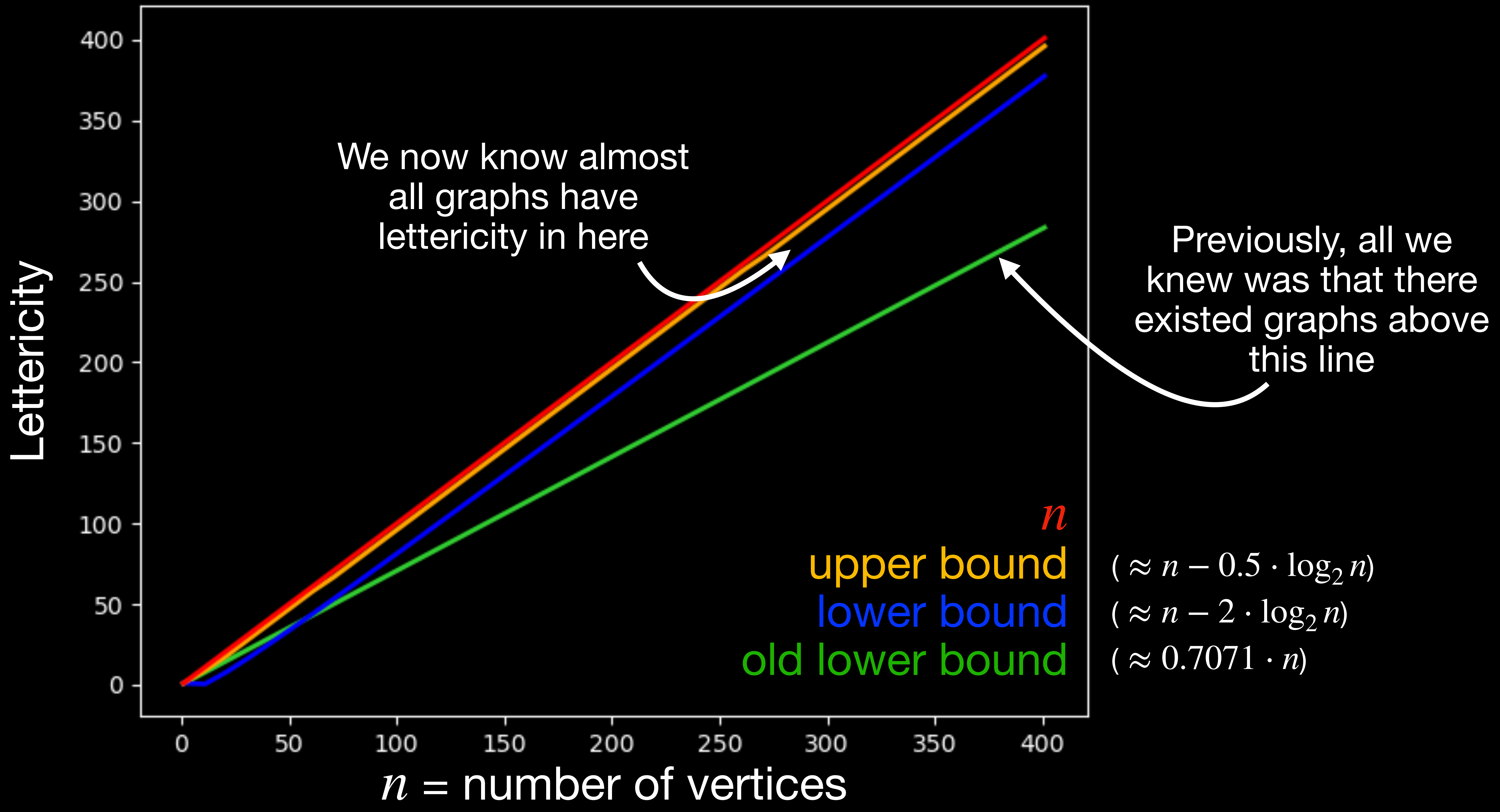
- **Theorem:** For almost all graphs G on n vertices, we have

$$\ell(G) \geq n - (2\log_2 n + 2\log_2\log_2 n).$$

Proof: Since we have $\mathbb{P}\left[\bigcup C_{(v_i),\pi}\right] \leq (n)_{2k} \cdot k! \cdot 2^{-k(k-1)}$, plugging in $2\log_2 n + 2\log_2\log_2 n$ for k, we have $\mathbb{P}\left[\bigcup C_{(v_i),\pi}\right]$ goes to 0 as $n \to \infty$.

The result then follows from this and facts 1 and 2. ∎

Summary of bounds on lettericity



$(\approx n - 0.5 \cdot \log_2 n)$

$(\approx n - 2 \cdot \log_2 n)$

$(\approx 0.7071 \cdot n)$

A natural question

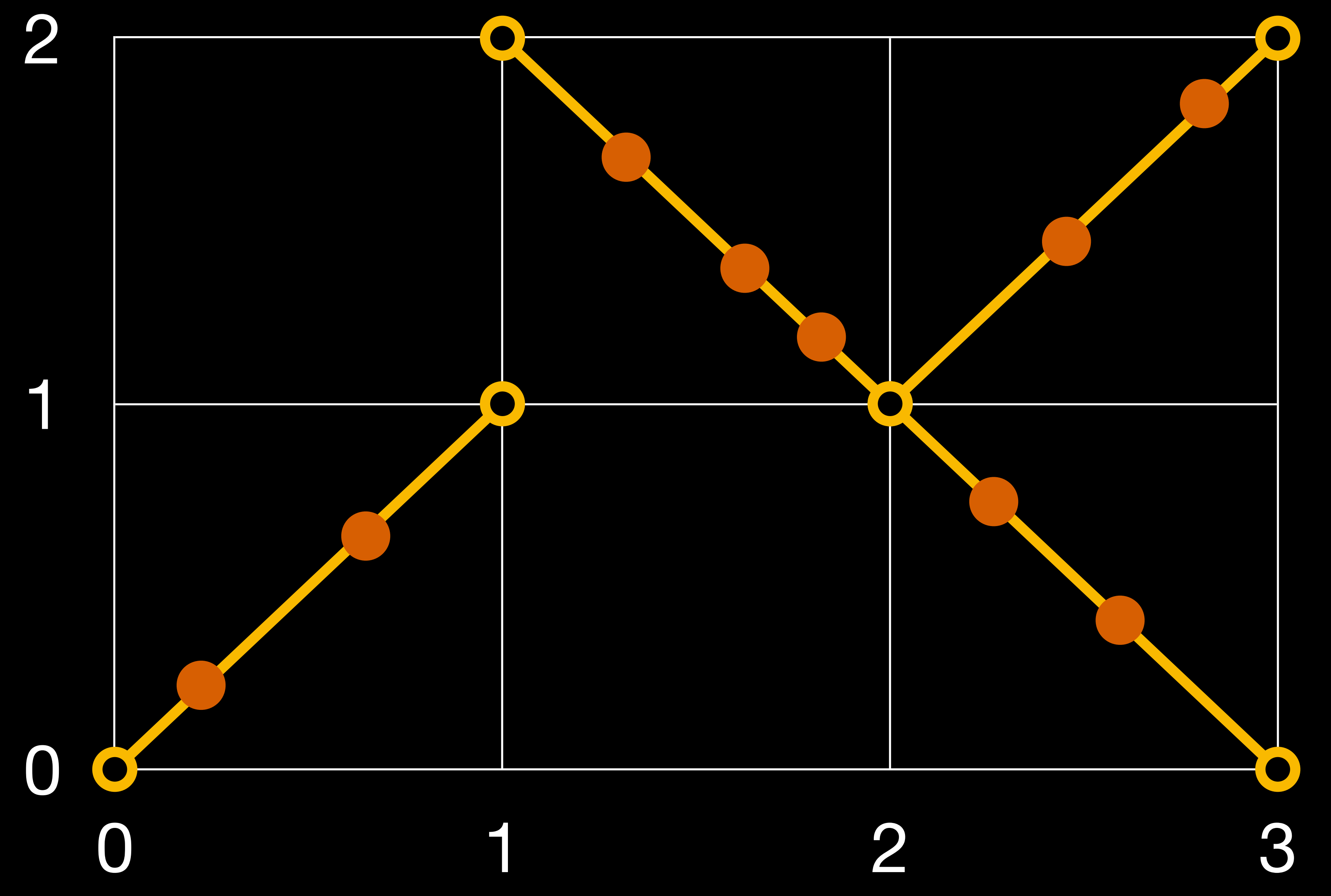
- ***Question:*** Can we obtain similar results pertaining to the maximal lettericity of inversion graphs?

- Because of the correspondence between lettericity and geometric grid classes, we can!

Returning to inversion graphs

- For our needs, the correspondence can be summarized in an example:

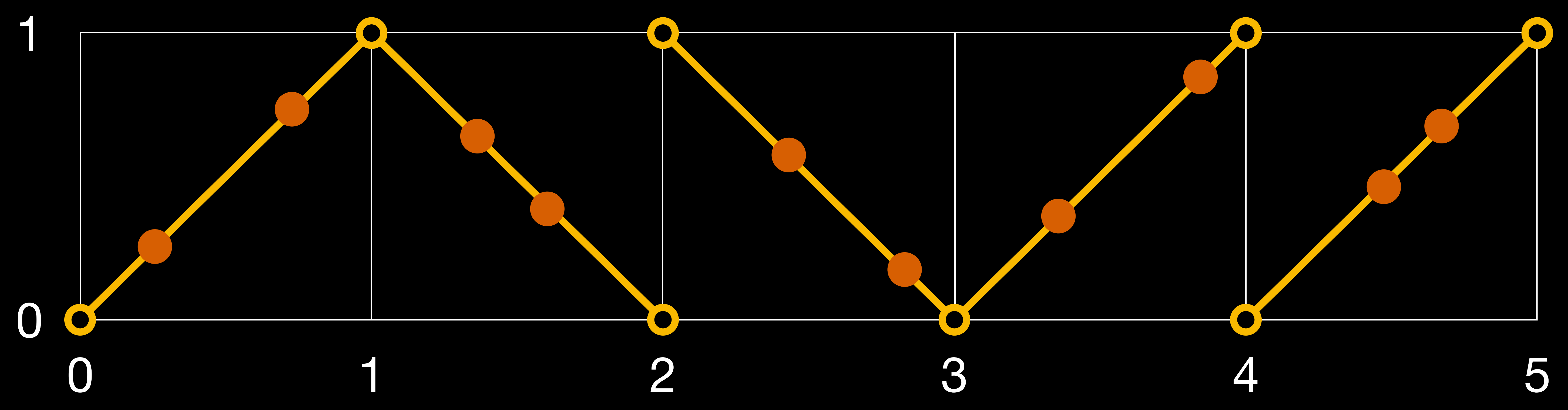
$$\pi \in \text{Geom}\begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \end{pmatrix} \implies \ell(G_\pi) \leq \sum_{i,j} |m_{i,j}| = 4$$



This drawing of a permutation on the standard figure maps to a lettering of its inversion graph in which the vertices in each cell are all encoded with the same letter.

Maximal lettericity of inversion graphs

- We can draw every permutation on the standard figure of a row ± 1matrix with at most $\lceil n/2 \rceil$ entries:

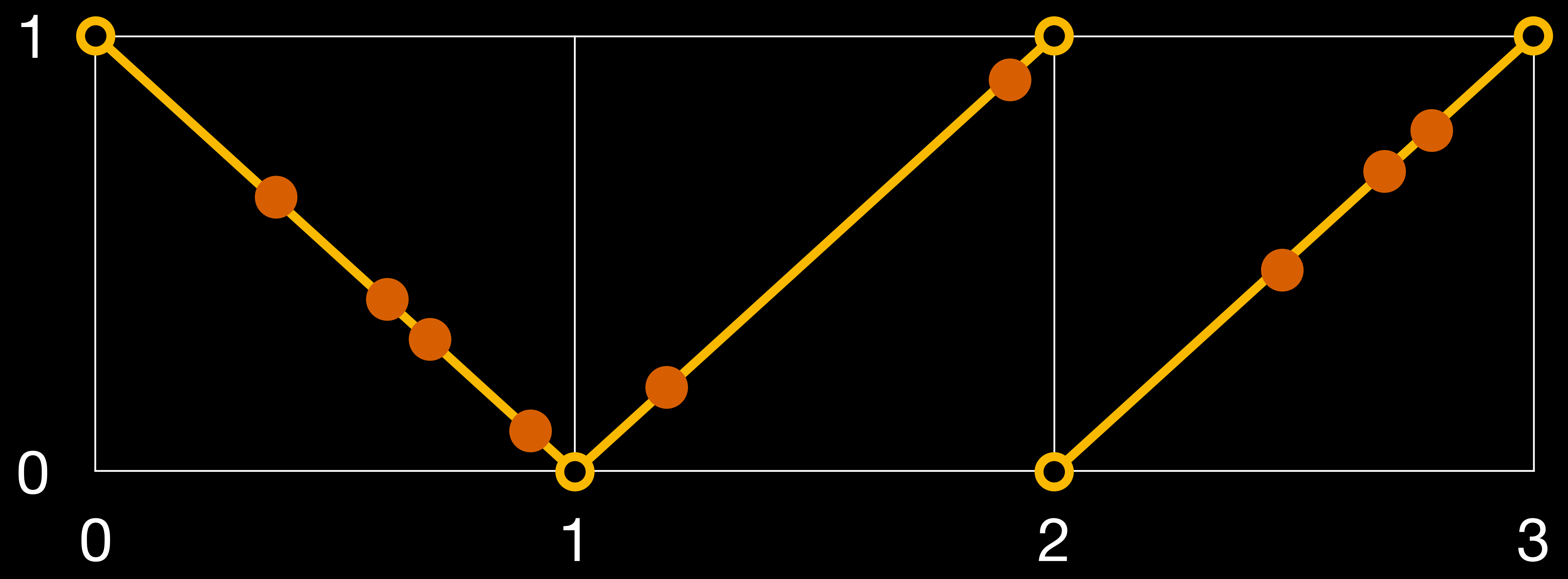


- Immediately, this shows that $\ell(G_\pi) \leq \lceil n/2 \rceil$ for all $\pi \in S_n$.

Expected lettericity of inversion graphs

- In using this row matrix approach, we can ask how many ‘bars’ do we need to write on a permutation so that the runs between the bars are monotone?

$$\pi = 6\,4\,3\,1 \;\Big|\; 2\,9 \;\Big|\; 5\,7\,8$$



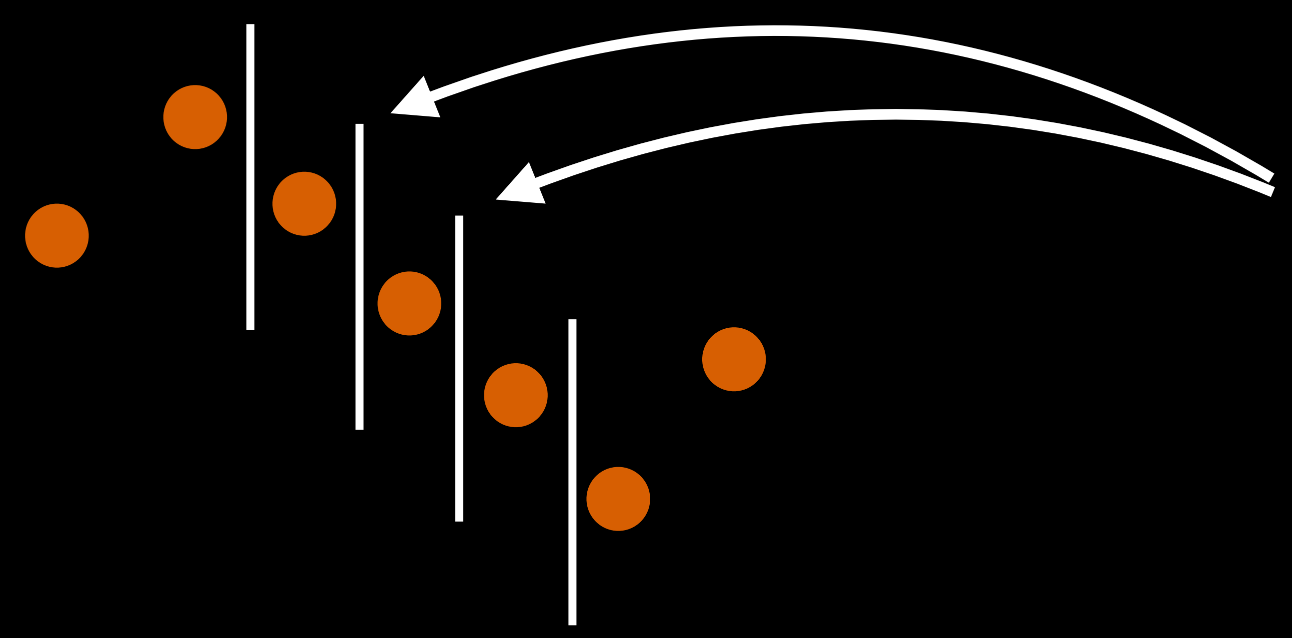
- Thus we have that $\ell(G_\pi) \leq 3$.

Expected lettericity of inversion graphs (2)

- By drawing a bar between the two entries of each descent, we partition a permutation into its ascending runs. It is known that

$$\mathbb{E}[\text{\# of ascending runs}] = \frac{n+1}{2}.$$

- We can, of course, delete a lot of these bars drawn at the descents:



We can delete these and still have the permutation partitioned into monotone runs.

- These bars correspond exactly to descending runs of length 4, so we can subtract off $\mathbb{E}[\text{\# of length 4 descending runs}] = (n-3)/24$.

Expected lettericity of inversion graphs (3)

- Using other similar pattern counting arguments, one can obtain

$$\mathbb{E}[\ell(G_\pi)] \leq 0.41n + C$$

for some constant C.

- (Note, this 0.41 can likely be lowered a bit further).

Part 2

Graphical Tranpositions

Transpositions in permutations

- For two integers $i, j \in [n]$, a ***transposition*** $T_{i,j}$ is a permutation given by the 2-cycle $(i \;\; j)$.

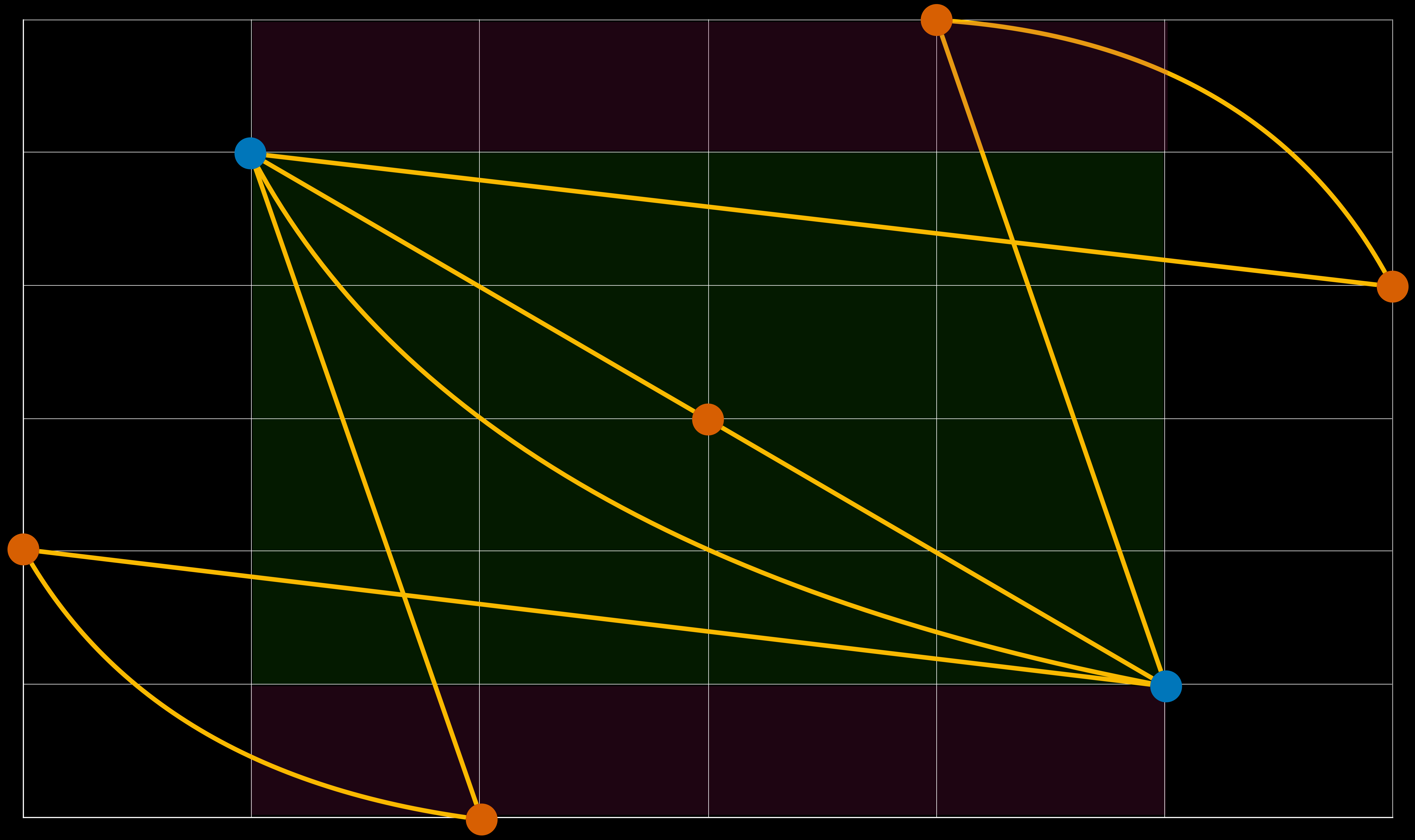
- For a permutation $\pi \in S_n$, the application of a transposition $T_{\pi(k),\pi(l)}$ on the left swaps the entries $\pi(k)$ and $\pi(l)$:

$$\pi = \quad \pi(1)\,\pi(2)\,\ldots\,\pi(k)\,\ldots\,\pi(l)\,\ldots\,\pi(n)$$

$$T_{\pi(k),\pi(l)} \circ \pi = \quad \pi(1)\,\pi(2)\,\ldots\,\pi(l)\,\ldots\,\pi(k)\,\ldots\,\pi(n)$$

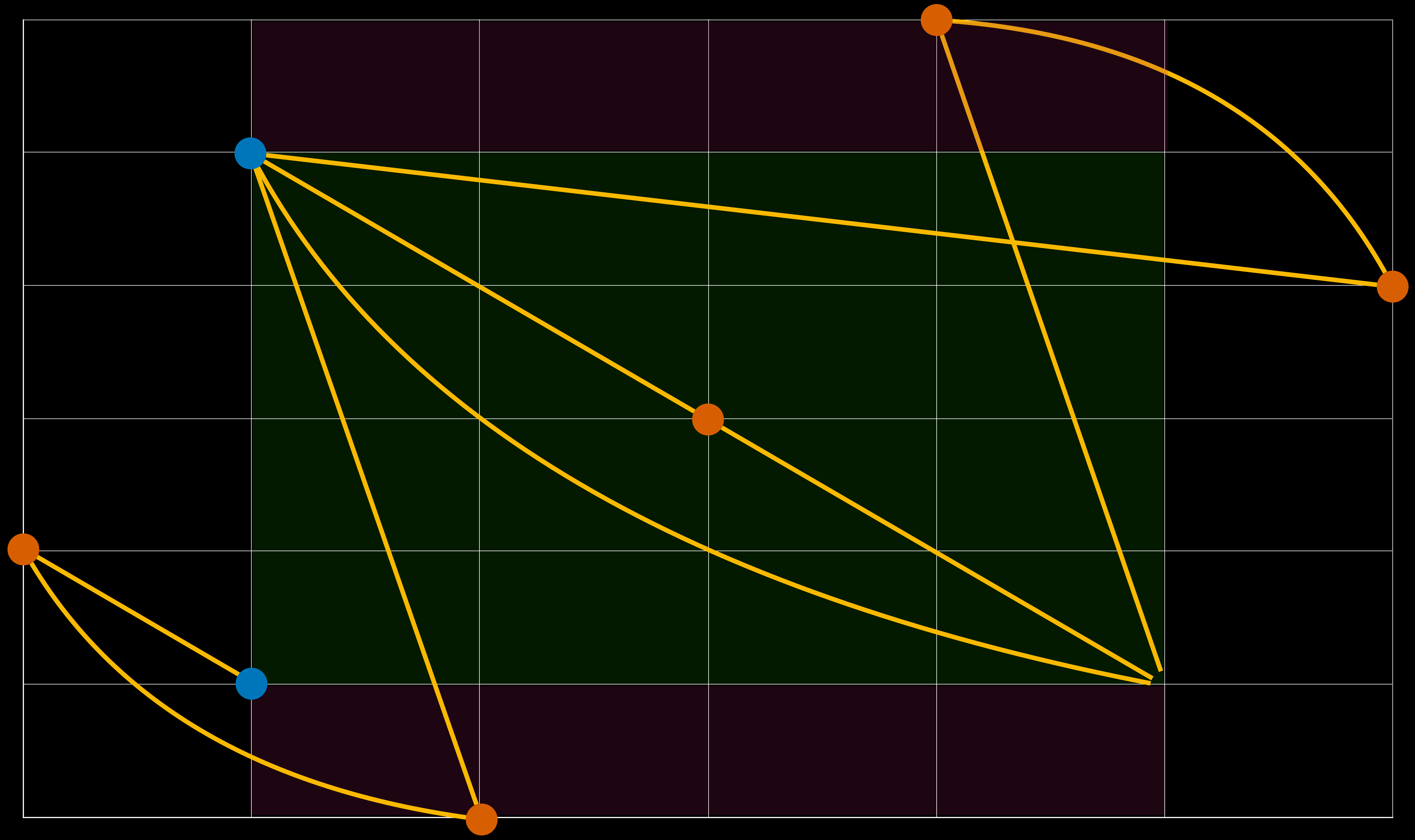
One-line notation

Transpositions and inversion graphs



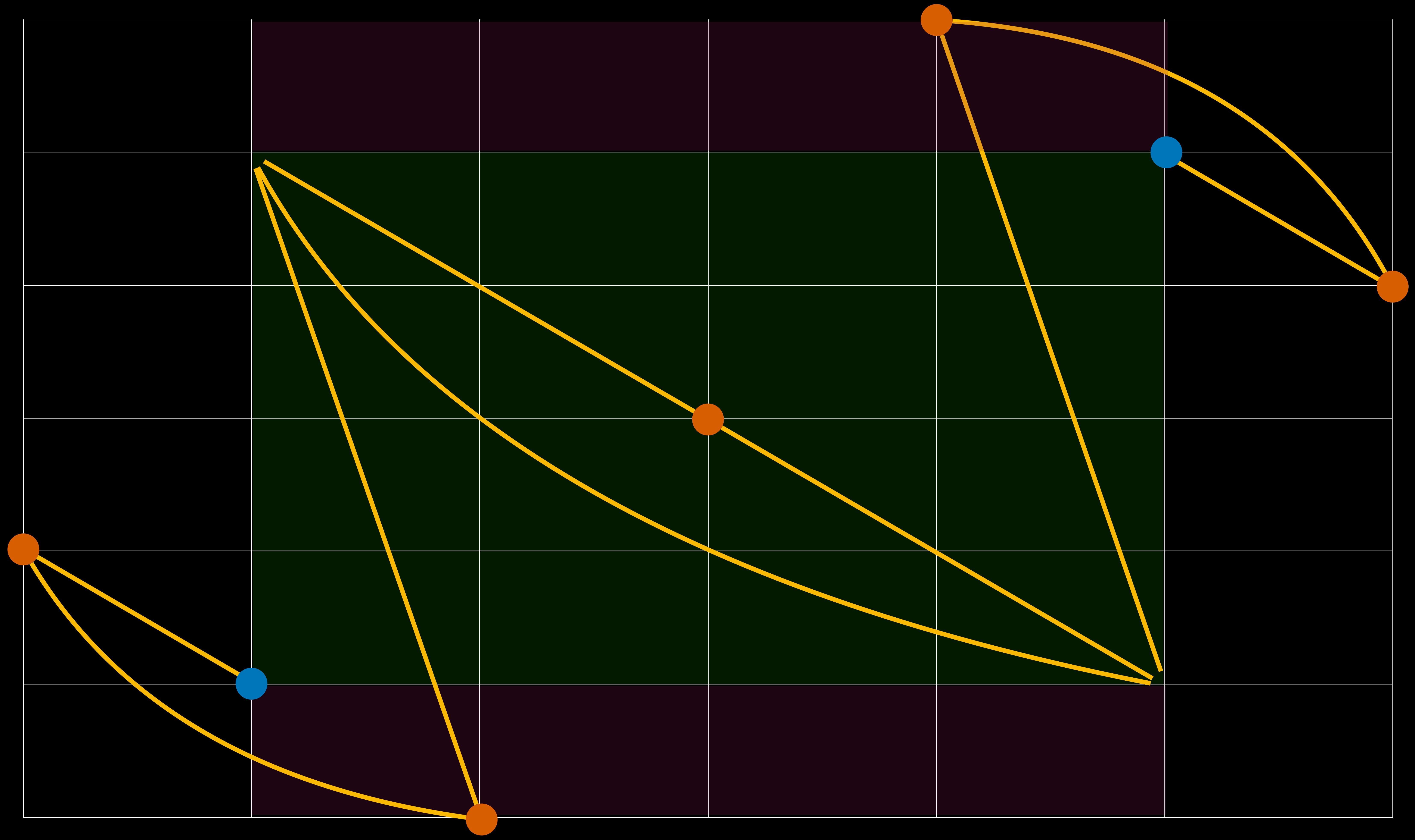
Applying $T_{2,6}$ to 3164725.

Transpositions and inversion graphs



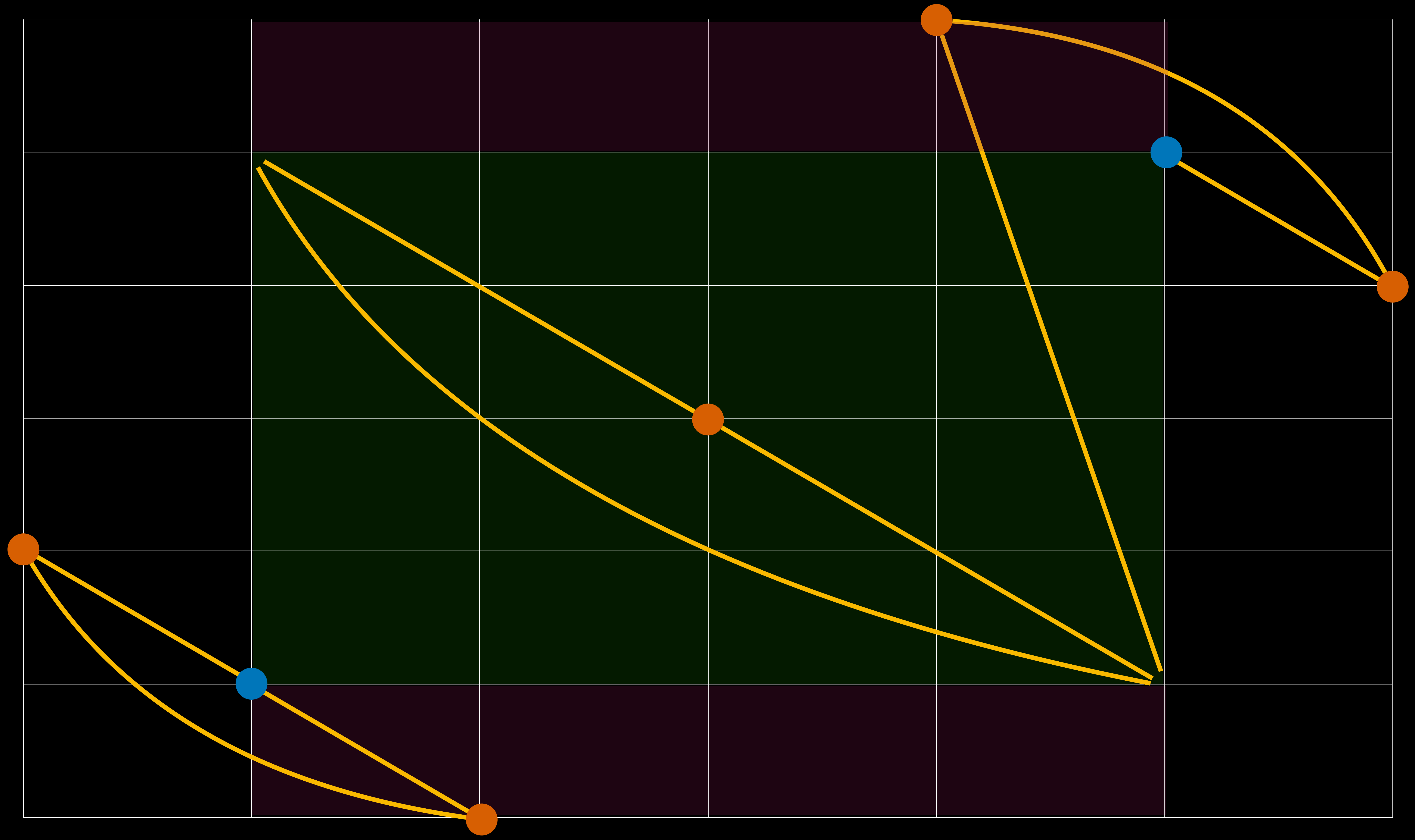
Applying $T_{2,6}$ to 3164725.

Transpositions and inversion graphs



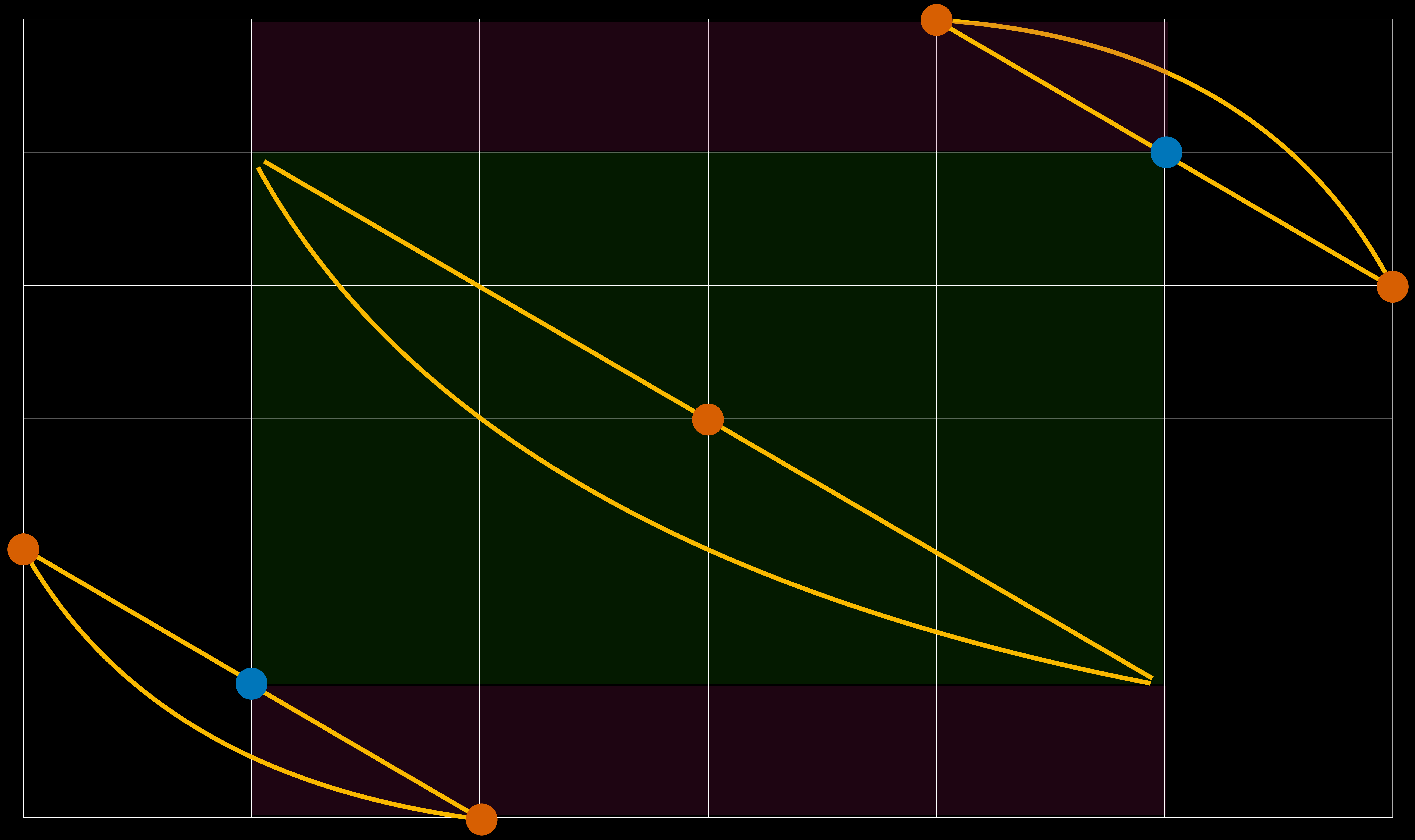
Applying $T_{2,6}$ to 3164725.

Transpositions and inversion graphs



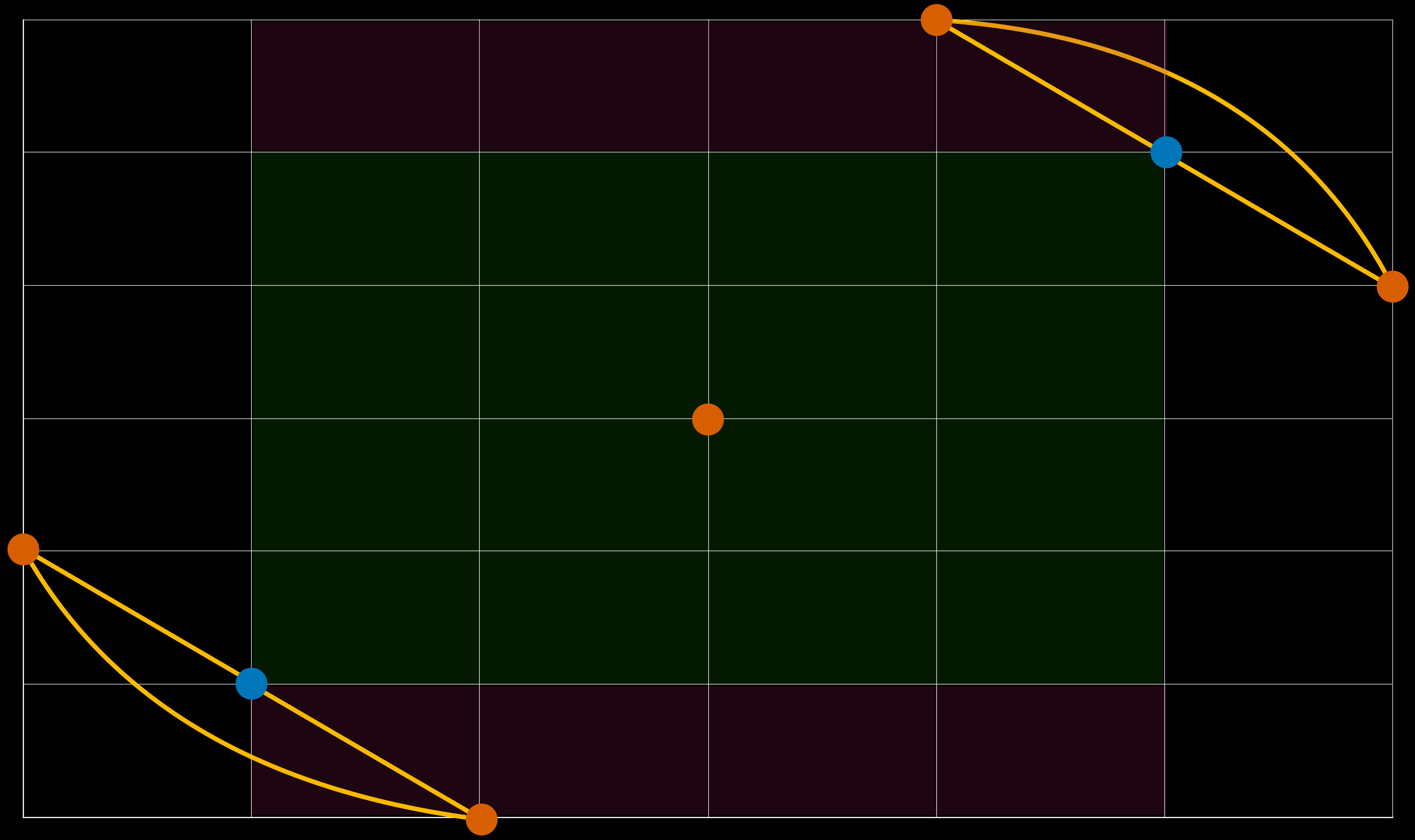
Applying $T_{2,6}$ to 3164725.

Transpositions and inversion graphs



Applying $T_{2,6}$ to 3164725.

Transpositions and inversion graphs



Applying $T_{2,6}$ to 3164725.

Transpositions and inversion graphs (2)

- In summary, by transposing the vertices (entries) of the edge (inversion) uv, we saw that:

1. Some of the nodes adjacent to exactly one of u and v were kept adjacent to that node, and some others were ‘passed’ to the other node.

2. For some of the nodes that were adjacent to both u and v, both the incident edges were deleted.

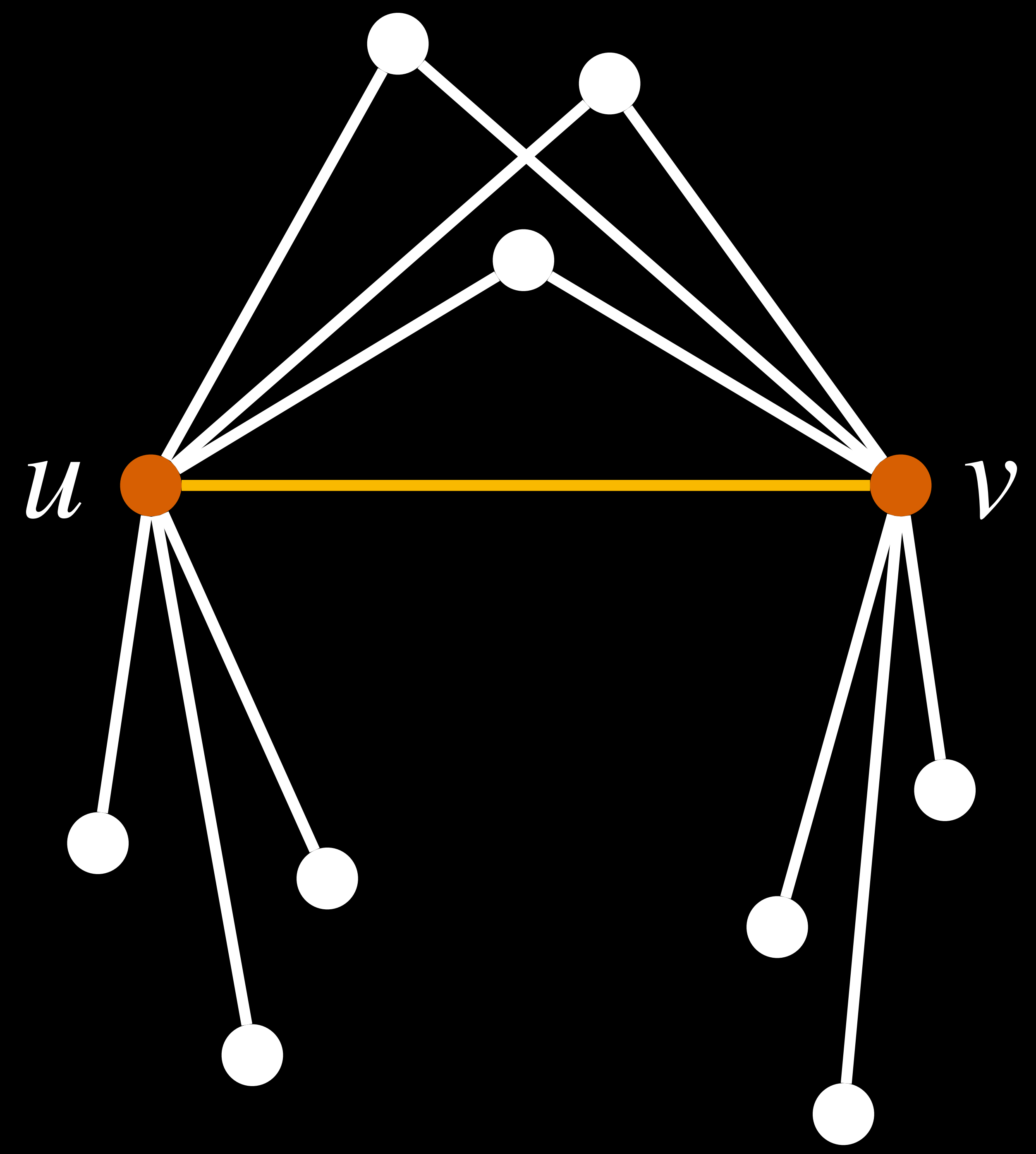
3. The edge uv was deleted.

Graphical transpositions

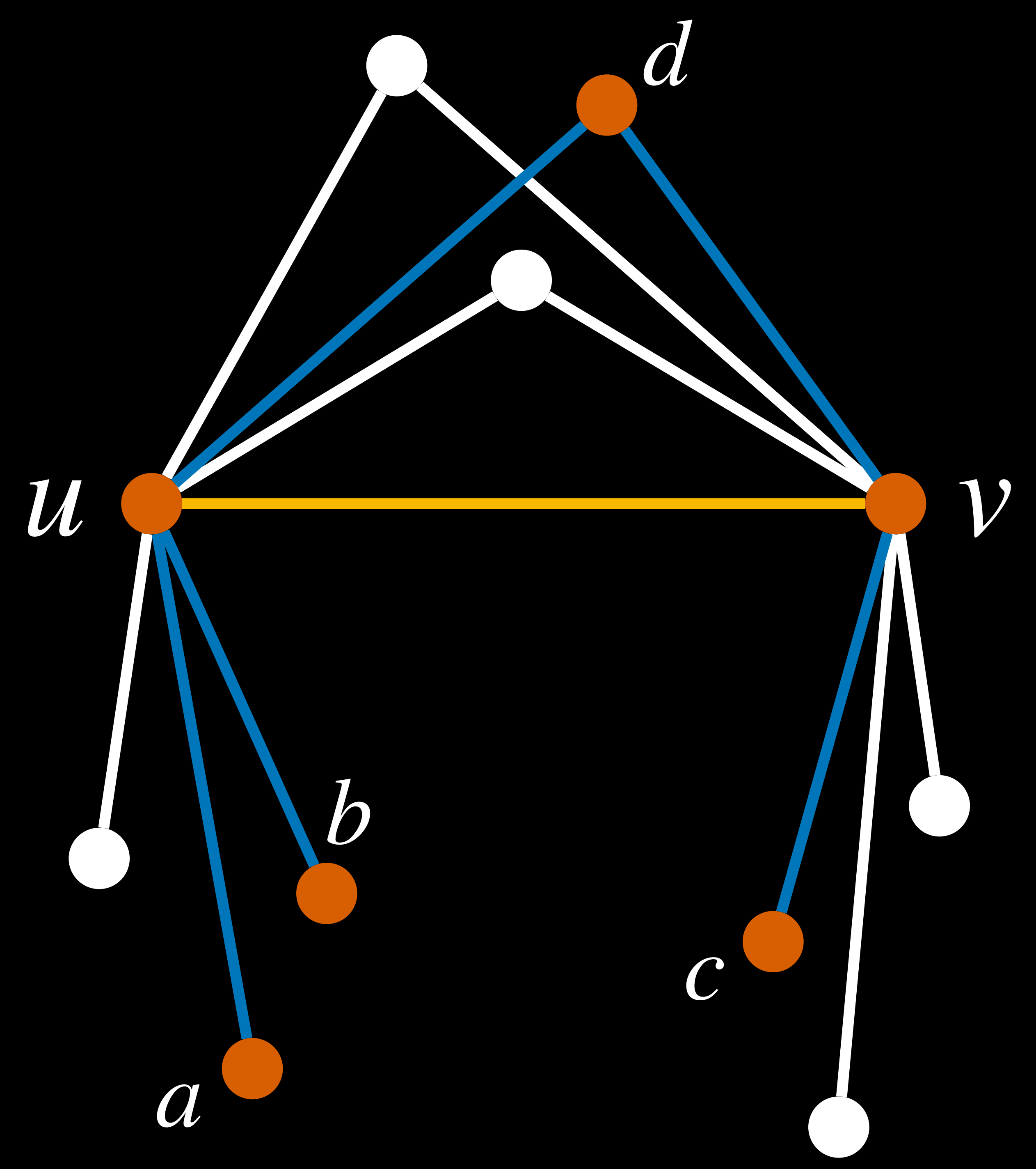
- These observations suggest the following graph operation.
- A ***transposition*** in a graph G consists of the following steps:
  1. Choose an edge $uv \in E(G)$.
  2. Choose a subset $X \subseteq N(u) \cup N(v)$ containing u and v.
  3. Toggle all edges between $\{u, v\}$ and X, including the edge uv.
- We will denote this transposition by T_{uv}^X.

Graphical transposition example

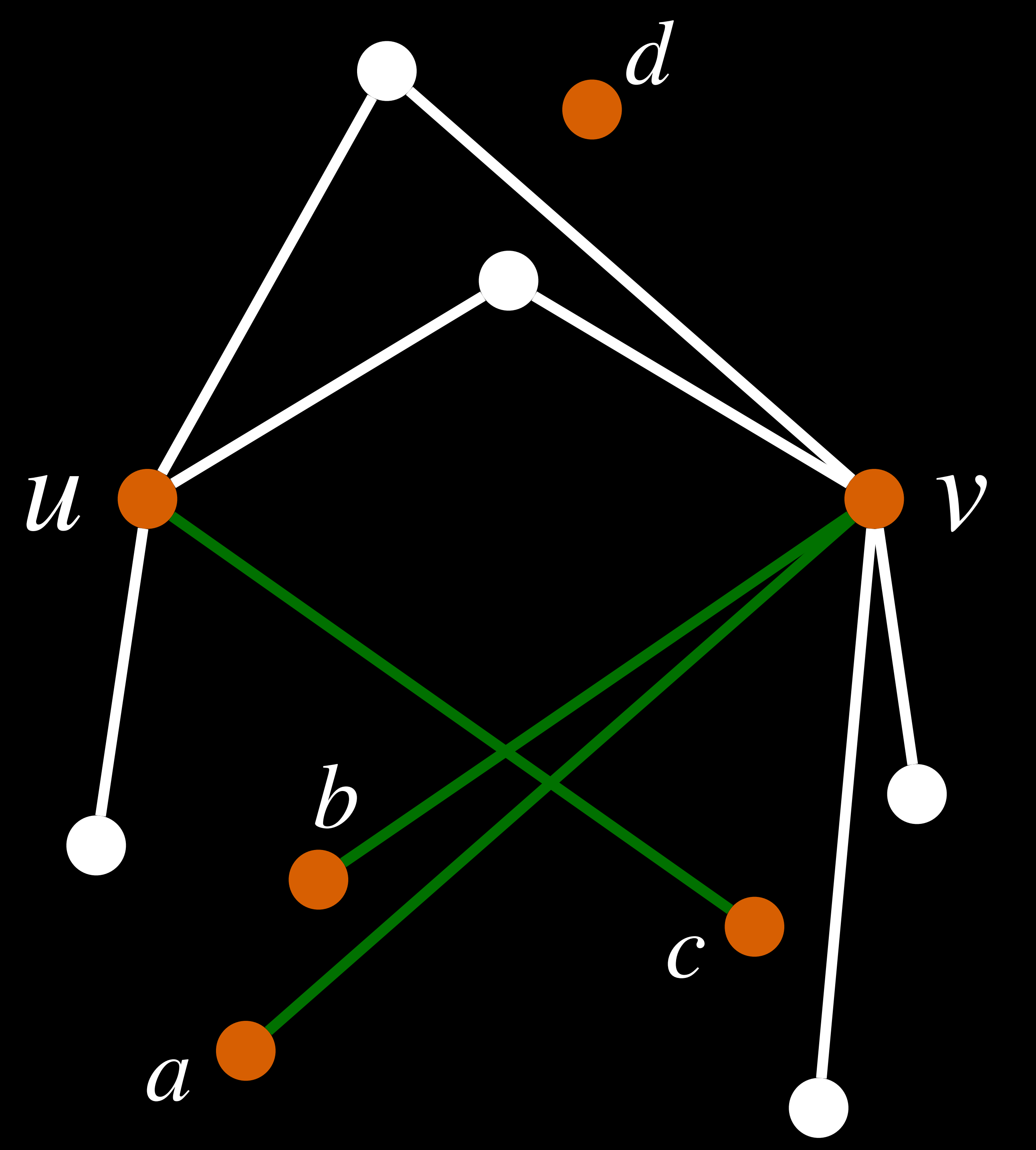
1. Choose an edge $uv \in E(G)$



2. Choose $X \subseteq N(u) \cup N(v)$



3. Toggle edges between $\{u, v\}$ and X



(Only edges incident to u and v are drawn)

Our motivation

- Using this operation, we ask:

1. Can we turn questions about permutations into interesting graph theory problems?

2. Might problems concerning permutations have simpler solutions from this graph-theoretic perspective?

Absolute length

- For a permutation $\pi \in S_n$, we will think of its ***absolute length*** as the least number of transpositions that can be applied to π to reach the identity permutation.
- Equivalently, the absolute length of π is the least k such that $\pi = t_1 t_2 \ldots t_k$ for transpositions t_i.
- ***Theorem:*** For a permutation $\pi \in S_n$ with c cycles, the absolute length of π is given by $n - c$.

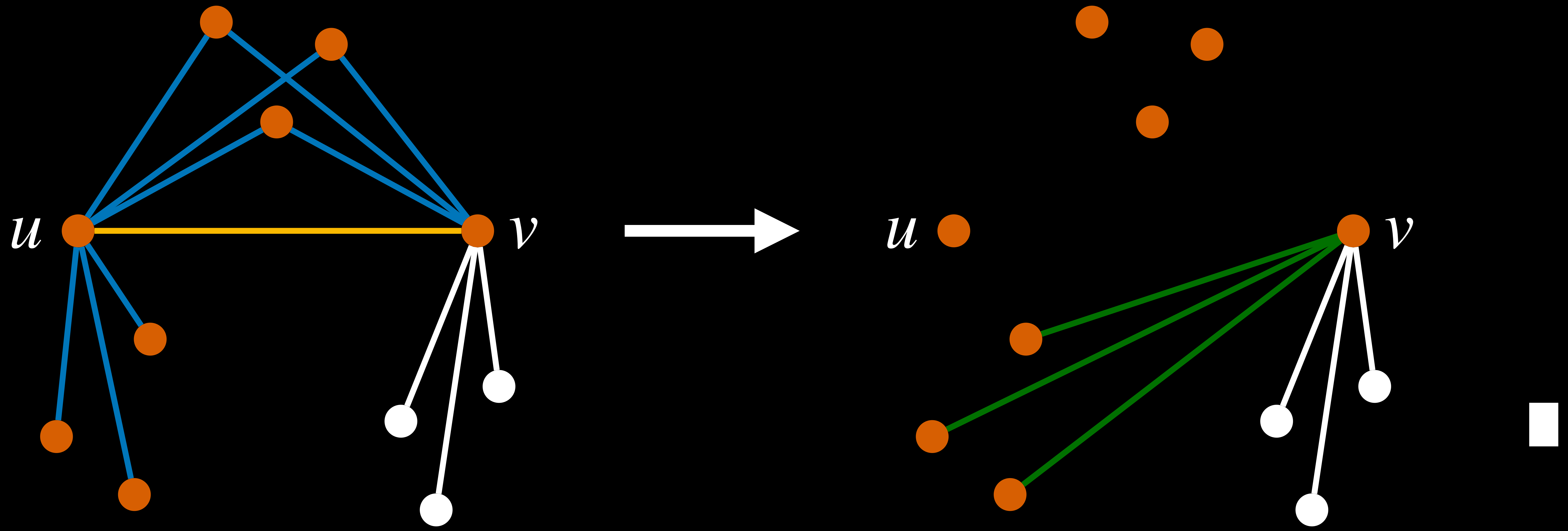
Proof: This follows from the fact that the absolute length of a k-cycle is $k - 1$:

$$(1\ 2\ 3\ \ldots\ k) = (1\ k)\ \ldots\ (1\ 3)(1\ 2)\,. \blacksquare$$

A first observation

- ***Fact:*** In a graph G, we can isolate any vertex with a single transposition.

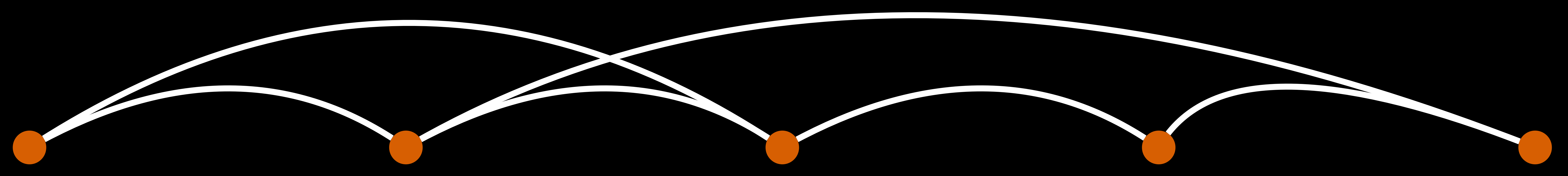
Proof: For any $u \in V(G)$, we can isolate it with the transposition $T_{uv}^{N[u]}$, where $N[u]$ is the closed neighborhood of u, and v is any neighbor of u.



A first theorem

- ***Theorem:*** ***A*** graph G can be transformed into the edgeless graph with $n-1$ or fewer transpositions.

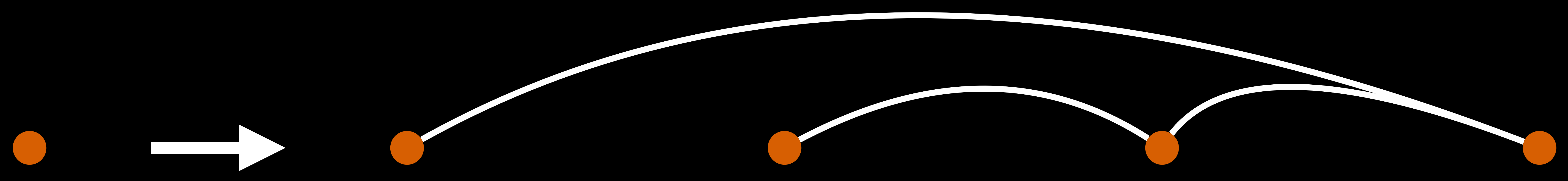
Proof: We repeatedly isolate at least one vertex with a transposition, and we always get the last vertex for free.



A first theorem

- ***Theorem:*** ***A*** graph G can be transformed into the edgeless graph with $n-1$ or fewer transpositions.

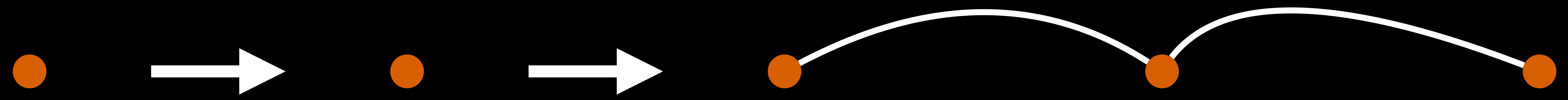
Proof: We repeatedly isolate at least one vertex with a transposition, and we always get the last vertex for free.



A first theorem

- ***Theorem:*** ***A*** graph G can be transformed into the edgeless graph with $n-1$ or fewer transpositions.

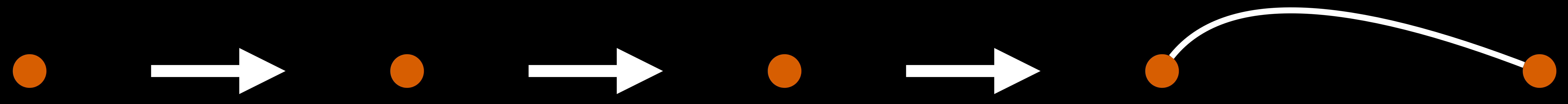
  Proof: We repeatedly isolate at least one vertex with a transposition, and we always get the last vertex for free.



A first theorem

- ***Theorem:*** ***A*** graph G can be transformed into the edgeless graph with $n - 1$ or fewer transpositions.

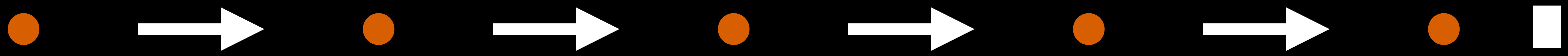
Proof: We repeatedly isolate at least one vertex with a transposition, and we always get the last vertex for free.



A first theorem

- ***Theorem:*** ***A*** graph G can be transformed into the edgeless graph with $n-1$ or fewer transpositions.

  Proof: We repeatedly isolate at least one vertex with a transposition, and we always get the last vertex for free.

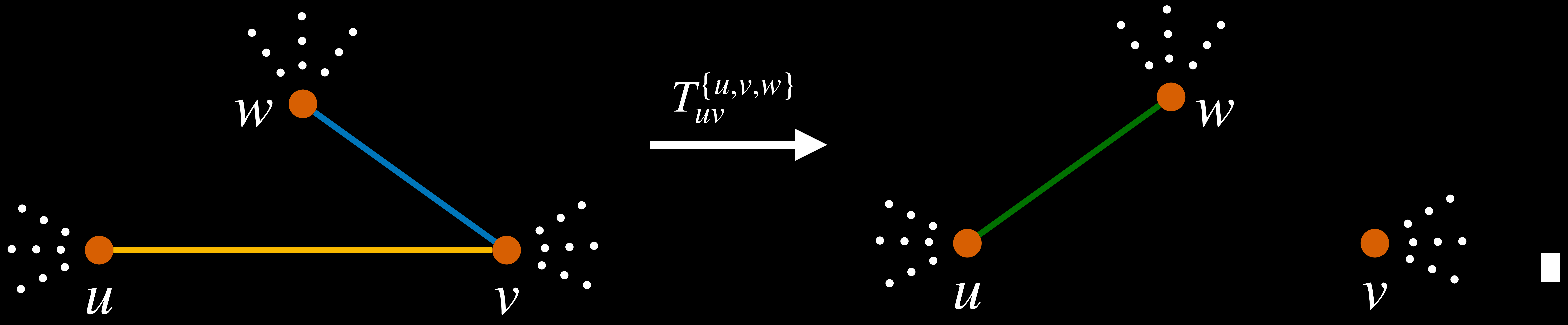


Extremal graphs

- So for which graphs are these bounds tight to the minimum number of transpositions needed to reach the edgeless graph?

- ***Proposition:*** Applying any single transposition to a forest results in a forest with one fewer edge.

Proof: In applying any transposition T_{uv}^{X} to a forest, only the edge uv is deleted, and no cycles can be formed.



Extremal graphs (2)

- ***Corollary:*** For a tree on n vertices, exactly $n - 1$ transpositions are required to transform it into the edgeless graph.

- Further, for a forest with m edges, exactly m transpositions are required to transform it into the edgeless graph.

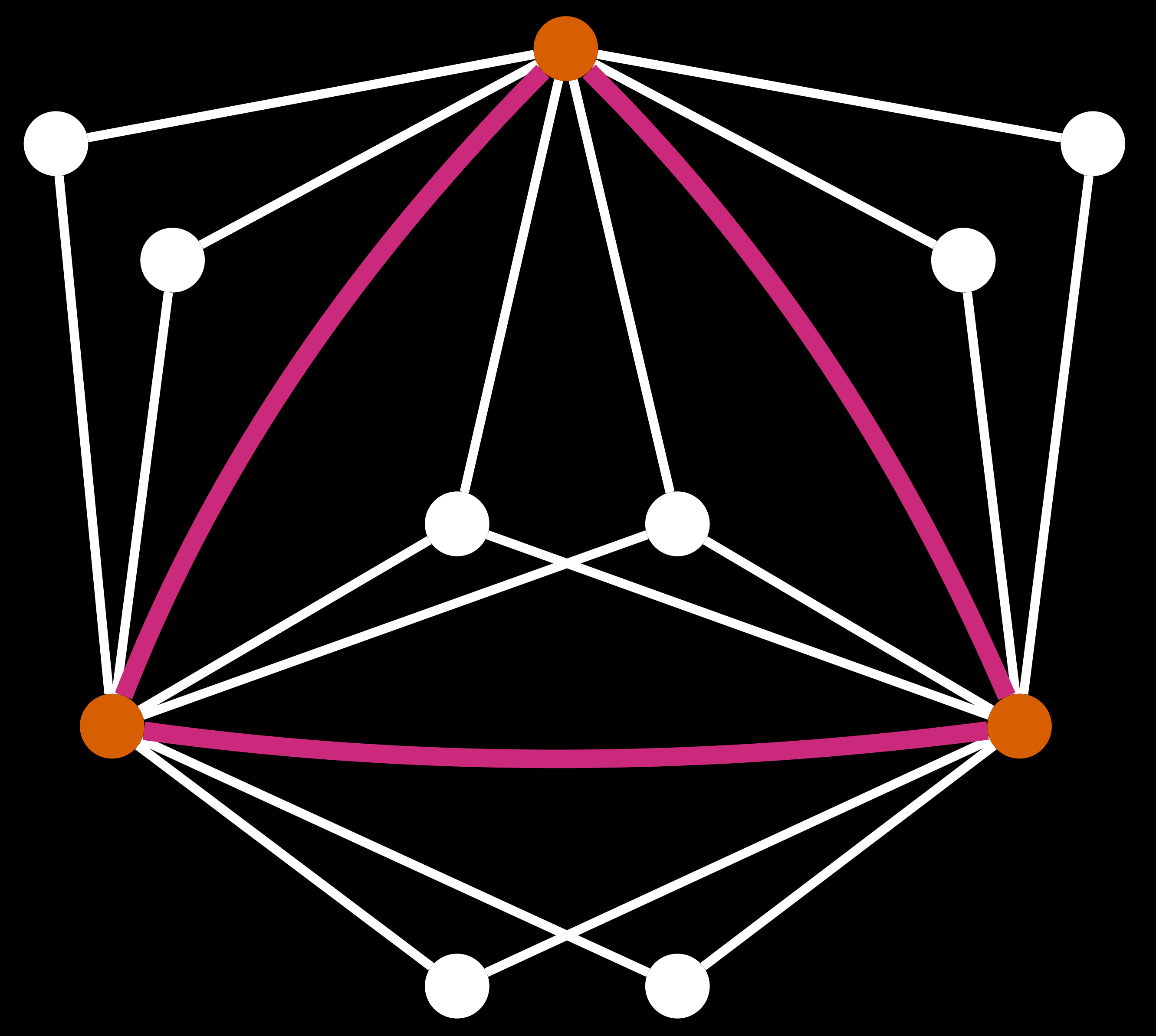
- But are there other extremal graphs?

Extremal graphs ***(3)***

- ***Theorem:*** A graph on n vertices requires exactly $n-1$ transpositions to reach the edgeless graph if and only if it is a tree.

Proof: All that remains to show is that if a graph on n vertices has a cycle, we can reach the edgeless graph with fewer than $n-1$ transpositions.

This takes a quite a bit of work. Essentially, we can reach a graph that looks like:

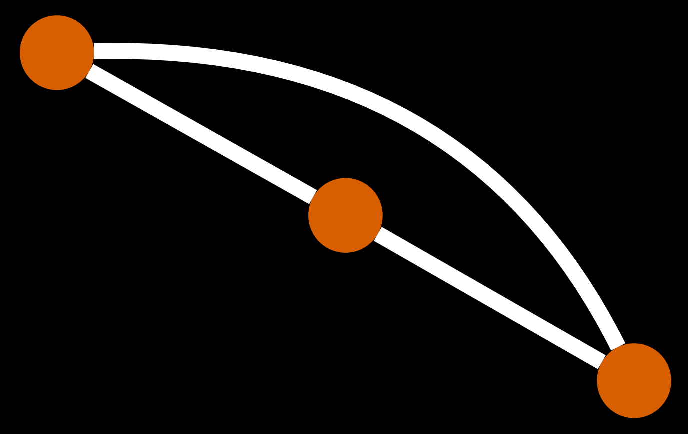
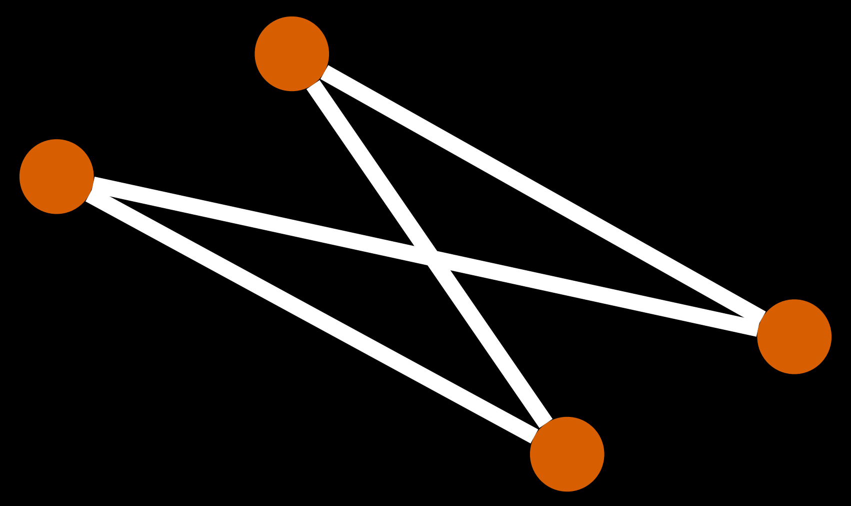


And then there are a couple cases to deal with. ∎

Returning to permutations

- The permutations whose inversions graphs are forests are referred to as ***boolean permutations***, and are exactly those that avoid both the patterns 321 and 3412.

  To see this, we note that 321 is the only length 3 pattern that yields a triangle, and 3412 is the only length 4 pattern that yields an induced 4-cycle:

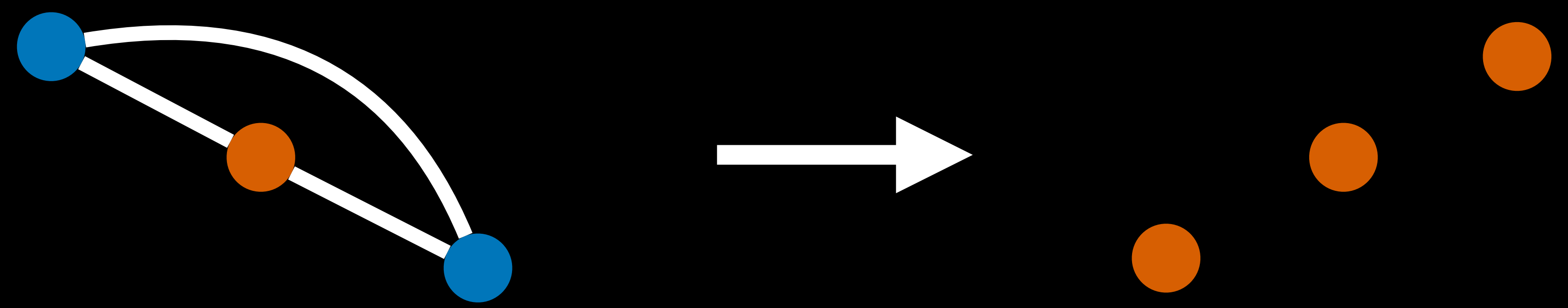
  Then, using some geometric reasoning in the plane, one can show that it is impossible to find a length 5+ pattern that yields an induced cycle as its inversion graph.

Returning to permutations (2)

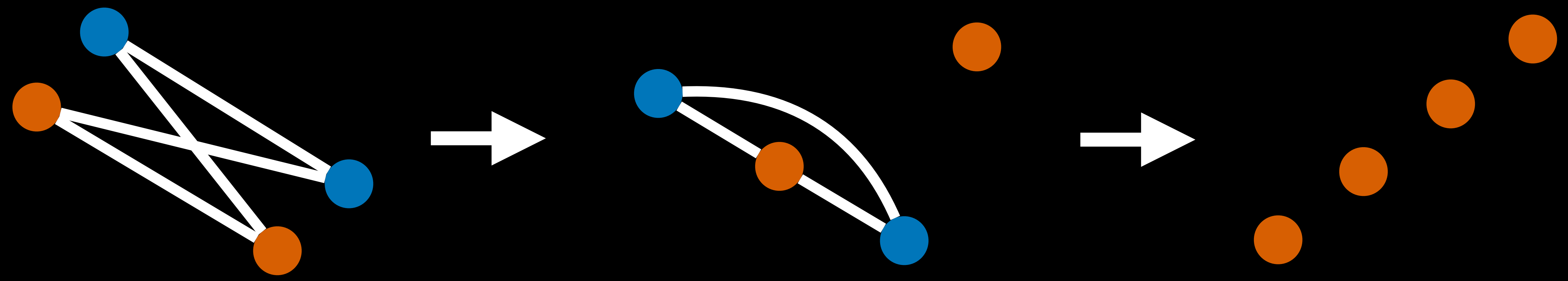
- ***Theorem:*** *(Edelman 1987, Petersen and Tenner 2014)* The length (number of inversions) and absolute length of a permutation coincide if and only if it is a boolean permutation.

  Proof: For a boolean permutation π, its inversion graph is a forest, and thus the correspondence of length and absolute length is clear from the graph-theoretic perspective.

  If π is not boolean then it contains either a 321 or 3412 pattern:



-3 edges with 1 transposition



-4 edges with 2 transpositions ∎

Some natural unanswered questions

- Can we find other bounds on the number of transpositions required to reach the edgeless graph in terms of other graph invariants?

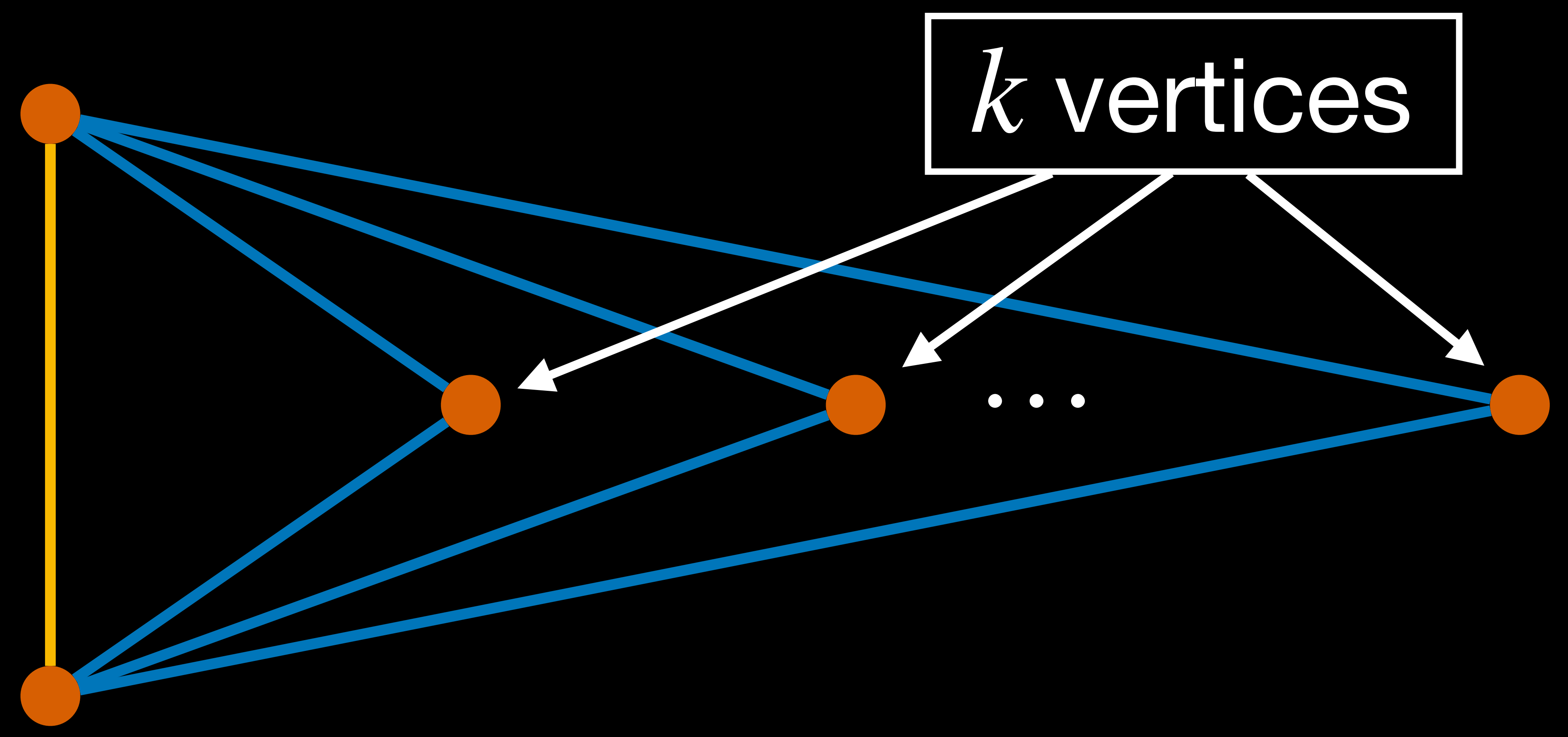
e.g. diameter, connectivity, girth, chromatic number, etc…

- Can we characterize the minimum number of transpositions required to reach the edgeless graph for some other well-known families of graphs?

e.g. complete k-partite graphs, threshold graphs, etc…

Graphs requiring one transposition

- It is not difficult to see that the connected graphs requiring exactly one transposition to reach the empty graph are given by

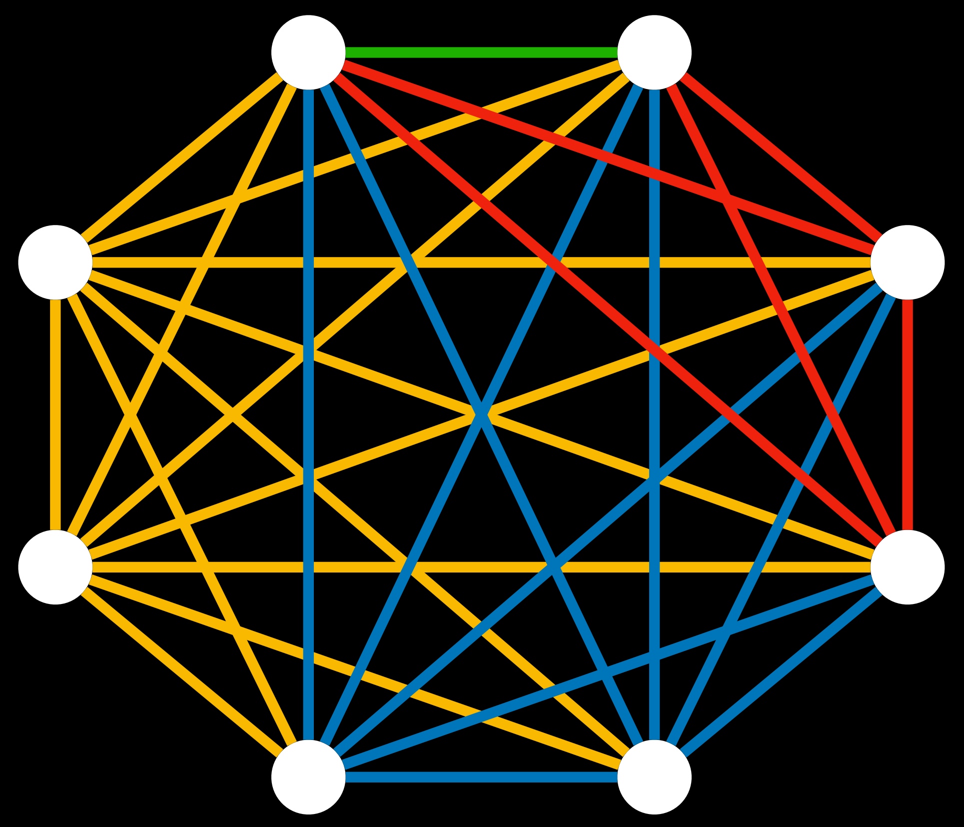
$N_k :=$ 

- We will call these graphs ***nested triangles***.
- Note that N_0 is the single edge graph K_2.

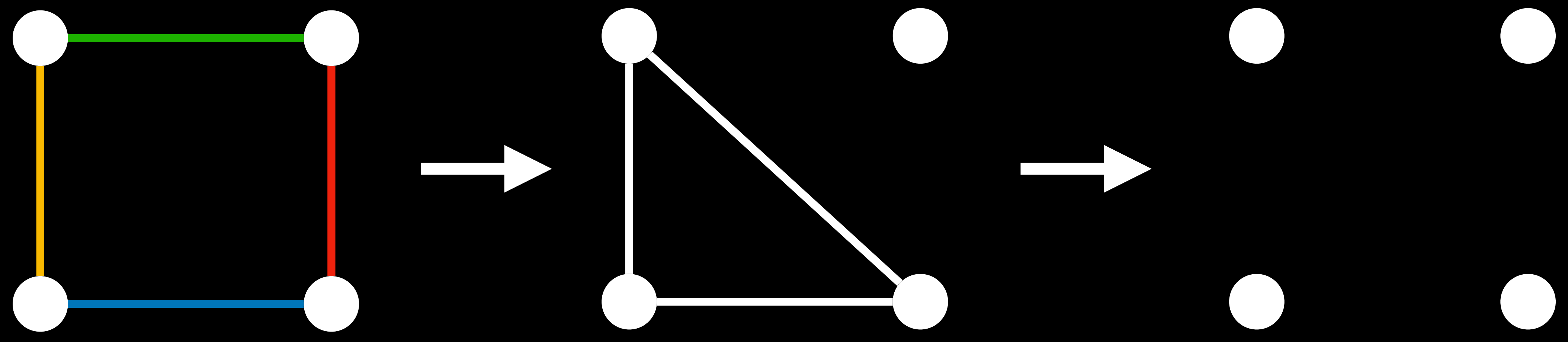
A problem

- ***Problem:*** For which graphs is it possible to consider only transpositions T_{uv}^{X} with $X \subseteq N(u) \cap N(v)$ so that the minimal number of these transpositions required to reach the edgeless graph is the same as with any transpositions?

- Equivalently, we can ask for which graphs do we only have to concern ourselves with partitioning its edges into nested triangles?



K_n is such a graph.

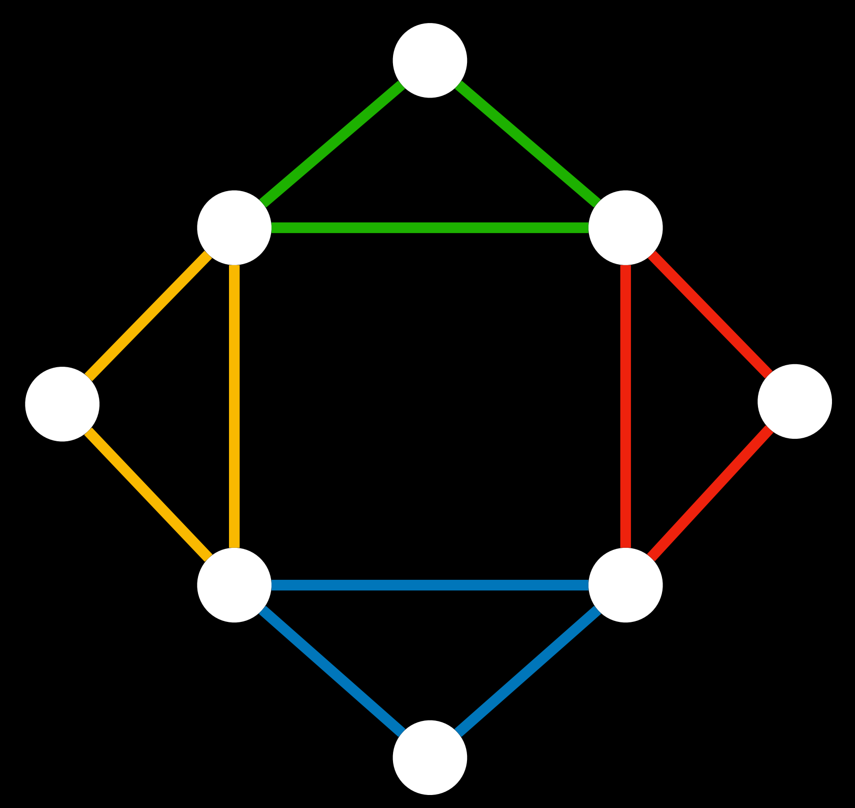


For $n \geq 4$, C_n is not.

A conjecture

- ***Conjecture:*** For the graphs in which all induced cycles are triangles, often referred to as ***chordal graphs***, we need only consider such transpositions.

- Note, these would not be the only such graphs.



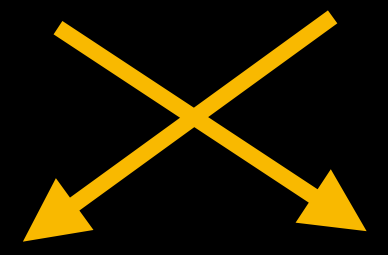
This graph has an induced 4-cycle, and requires four arbitrary transpositions to reach the empty graph, corresponding with the minimum size of a partition of its edges into nested triangles.

- After proving this conjecture, we would next ask: ‘can we relate this result to permutations that have chordal inversion graphs?’ These are 3412-avoiders.

Other future directions

- Can we generalize results concerning adjacent block transpositions of permutations to the setting of arbitrary simple graphs and obtain other interesting results?

$$\pi_1\ \pi_2\ \pi_3\ \pi_4\ \pi_5\ \pi_6\ \pi_7\ \pi_8\ \pi_9\ \pi_{10}\ \pi_{11}\ \pi_{12}$$



$$\pi_1\ \pi_2\ \pi_6\ \pi_7\ \pi_8\ \pi_9\ \pi_3\ \pi_4\ \pi_5\ \pi_{10}\ \pi_{11}\ \pi_{12}$$

- This operation would be equivalent to taking the symmetric difference with a complete bipartite graph on a subset of vertices.