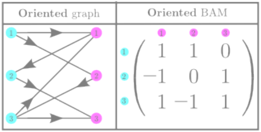
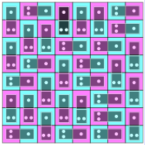
$$\det\begin{vmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{vmatrix} = 8$$

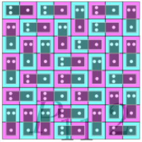
$$4^{16}\cdot\prod_{k=1}^{4}\prod_{\ell=1}^{4}\left(\cos^2\frac{k\pi}{9}+\cos^2\frac{\ell\pi}{9}\right) = 12,988,816$$



DETERMINANTS THAT COUNT

The Kastelyn method for dimer configurations and the Gessel-Viennot method for path systems

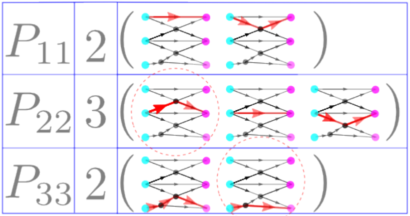


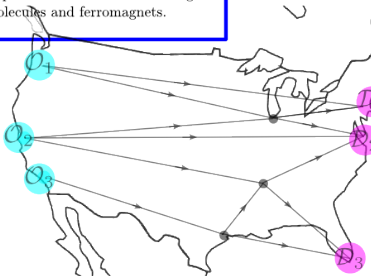


Homer Reid

Wednesday 1/23/2013 1:00 PM 2-190

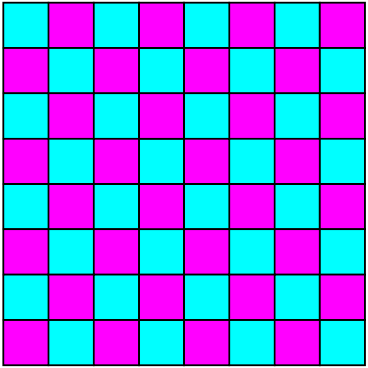
How many ways can you tile a chessboard with dominoes? How many ways can 3 salespeople visit 9 cities without overlapping? How many ways can a grid of microscopic magnets align or misalign with one another? Amazingly, all of these questions can be answered by writing down a simple matrix of integers and computing its determinant. In this lecture for 18.095 (MIT's IAP mathematics lecture series) we will learn these powerful methods of counting and touch briefly on their relation to the physics of diatomic molecules and ferromagnets.





Invitation: How many ways to tile a chessboard?

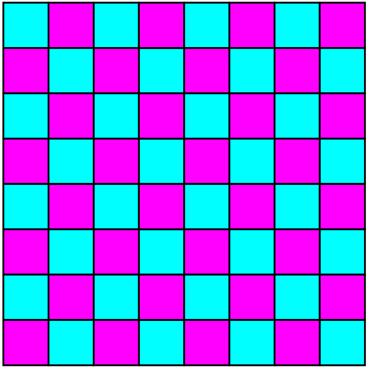
I give you a standard **8×8 chessboard...**

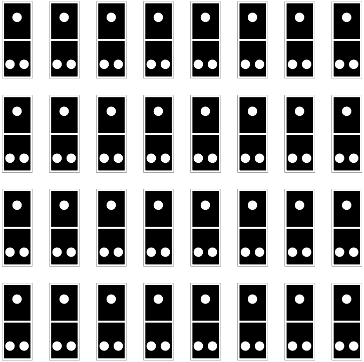


Invitation: How many ways to tile a chessboard?

I give you a standard **8×8 chessboard...**

...and 32 standard **2×1 dominoes...**

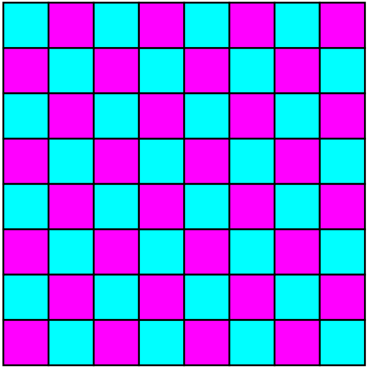


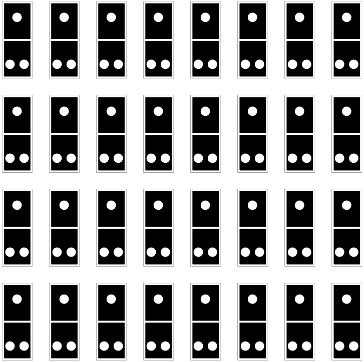


Invitation: How many ways to tile a chessboard?

I give you a standard **8×8 chessboard...**

...and 32 standard **2×1 dominoes...**



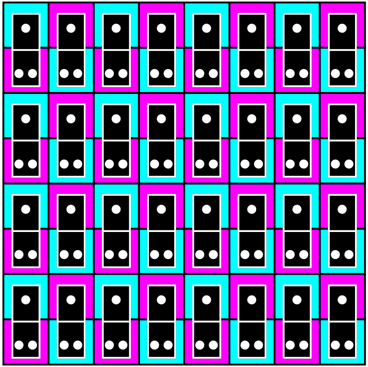


...and I ask: **How many ways can you cover the chessboard with the dominoes?**

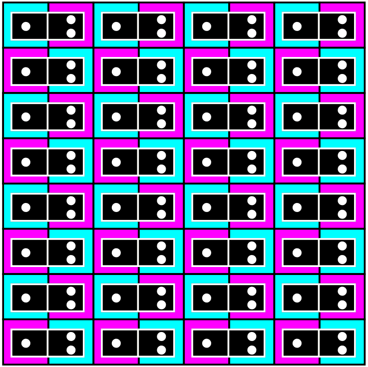
Invitation: How many ways to tile a chessboard?

Examples of tilings

Well, all the dominoes could be vertical...



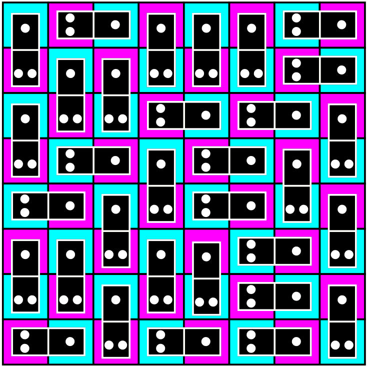
...or they could all be horizontal...



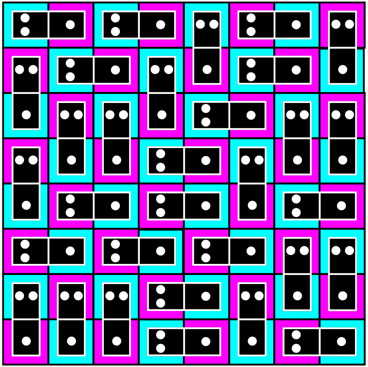
Invitation: How many ways to tile a chessboard?

Examples of tilings

or they could be all mixed up...



...or all mixed up a different way...

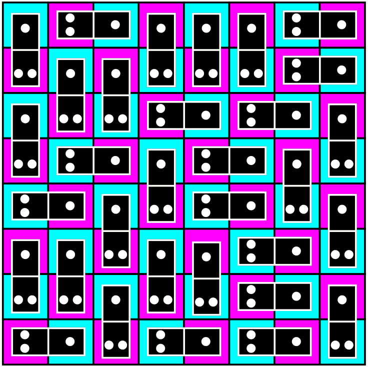


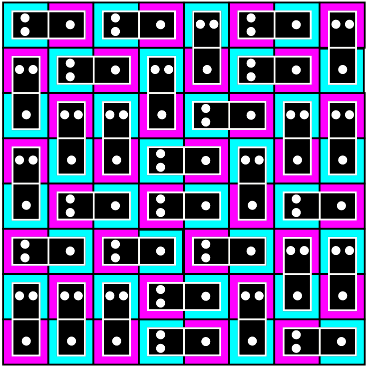
Invitation: How many ways to tile a chessboard?

Examples of tilings

or they could be all mixed up...

...or all mixed up a different way...





Clearly enumerating tilings by hand is hopeless...

...but there is an easier way.

Invitation: How many ways to tile a chessboard?

A sneak peek at the answer...

Astonishing fact:

The number of domino tilings of the 8×8 chessboard is **12,988,816.**

Invitation: How many ways to tile a chessboard?

A sneak peek at the answer...

Astonishing fact:

The number of domino tilings of the 8× 8 chessboard is **12,988,816.**

Even more astonishing fact:

This integer is computed as the product of 16 mostly irrational numbers:

$$\boxed{\texttt{Tilings}(8,8) = 4^{16} \cdot \prod_{k=1}^{4} \prod_{l=1}^{4} \left(\cos^2 \frac{k\pi}{9} + \cos^2 \frac{l\pi}{9} \right).}$$

Invitation: How many ways to tile a chessboard?

A sneak peek at the answer...

Astonishing fact:

The number of domino tilings of the 8× 8 chessboard is **12,988,816.**

Even more astonishing fact:

This integer is computed as the product of 16 mostly irrational numbers:

$$\boxed{\texttt{Tilings}(8,8) = 4^{16}\cdot\prod_{k=1}^{4}\prod_{l=1}^{4}\left(\cos^2\frac{k\pi}{9}+\cos^2\frac{l\pi}{9}\right).}$$

More generally, for an $M\times N$ chessboard (with M,N even:)

$$\boxed{\texttt{Tilings}(M,N) = 4^{\frac{MN}{4}}\cdot\prod_{k=1}^{\frac{M}{2}}\prod_{l=1}^{\frac{N}{2}}\left(\cos^2\frac{k\pi}{M+1}+\cos^2\frac{l\pi}{N+1}\right).}$$

Invitation: How many ways to tile a chessboard?

A sneak peek at the answer...

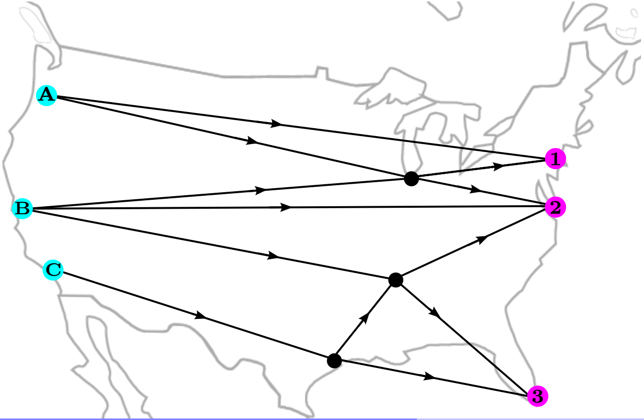
Perhaps **most astonishing of all:** 12,988,816 is a **32×32 determinant!**

$$12{,}988{,}816=\det\begin{vmatrix}
1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
-1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&-1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&-1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
-1&0&0&0&1&-1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&-1&0&0&0&1&-1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&-1&0&0&0&1&-1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&-1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&-1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&-1&0&0&-1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&-1&0&0&-1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&-1&0&0&-1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&-1&0&0&0&1&-1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&-1&0&0&0&1&-1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&-1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&-1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&-1&1&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&-1&1&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&-1&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&-1&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&-1&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&-1&1&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&-1&1&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&-1&1&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&-1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&-1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&-1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1
\end{vmatrix}$$

Invitation: How many ways to crisscross the country?

Counting vertex-disjoint lattice path systems

3 traveling salespeople start in 3 origin cities (A,B,C). How many different ways can they travel to 3 destination cities (1,2,3) **without overlapping at any city?**

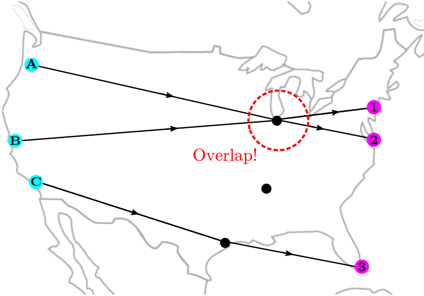


Invitation: How many ways to crisscross the country?

Counting vertex-disjoint lattice path systems

Non-overlapping (**"vertex-disjoint"**) vs. overlapping path systems:

OK (Vertex disjoint)

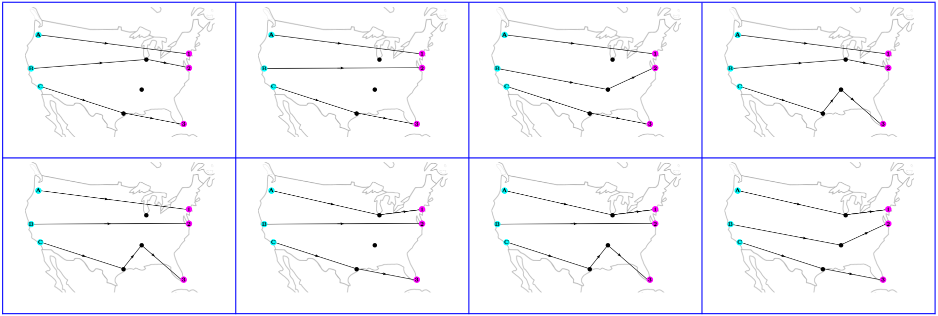


No good! **Not vertex disjoint.**

Invitation: How many ways to crisscross the country?

Counting vertex-disjoint lattice path systems

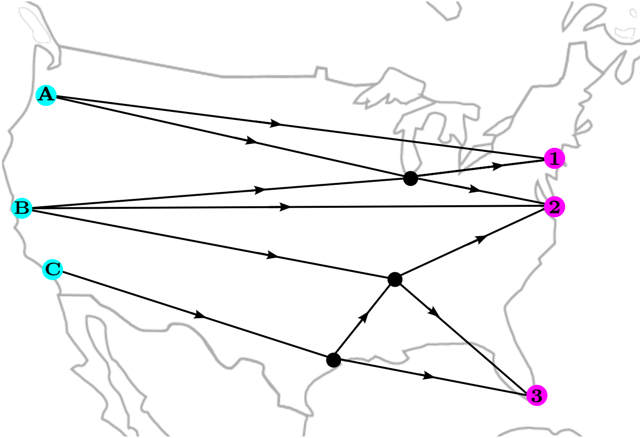
For this simple graph there are 8 vertex-disjoint path systems, which we can count by hand...



Invitation: How many ways to crisscross the country?

Counting vertex-disjoint lattice path systems

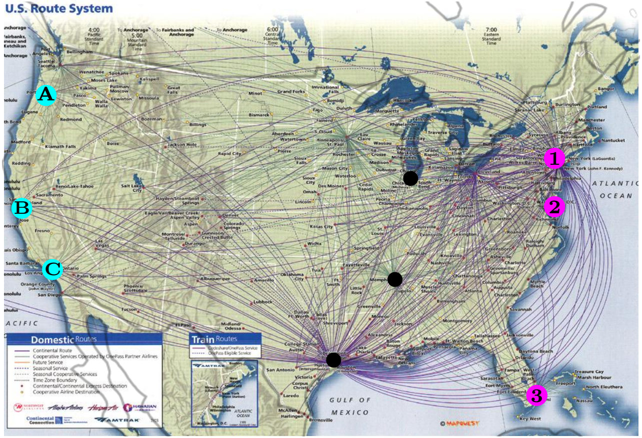
So it's easy enough to solve the problem this network...



Invitation: How many ways to crisscross the country?

Counting vertex-disjoint lattice path systems

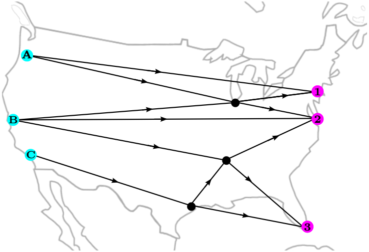
...but how about for this network?

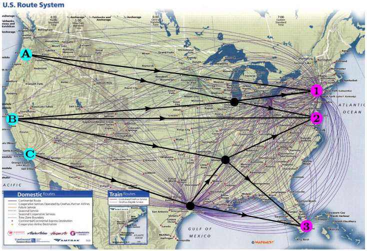


Invitation: How many ways to crisscross the country?

Counting vertex-disjoint lattice paths systems

Astonishing fact: **vertex-disjoint path systems are counted by a determinant.**





$$\det\begin{vmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{vmatrix} = 8$$

$$\det\begin{vmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{vmatrix} = ?$$

Determinants That Count

An Outline of this Talk

1. Determinants that count **domino tilings**:
   The Kastelyn-Percus-Hurst-Green Method

2. Determinants that count **lattice path systems**:
   The Gessel-Viennot Method

Determinants That Count

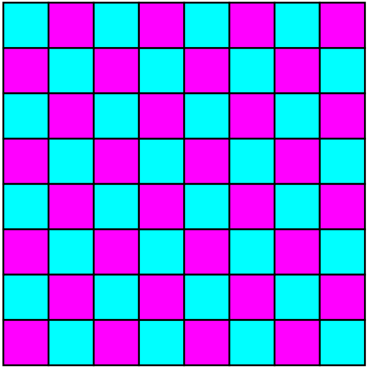
An Outline of this Talk

1. Determinants that count **domino tilings**:
   The Kastelyn-Percus-Hurst-Green Method

2. Determinants that count **lattice path systems**:
   The Gessel-Viennot Method

Invitation: How many ways to tile a chessboard?

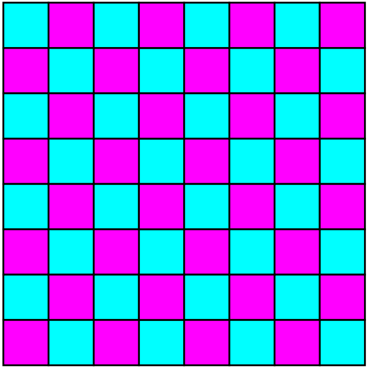
I give you a standard **8×8 chessboard...**

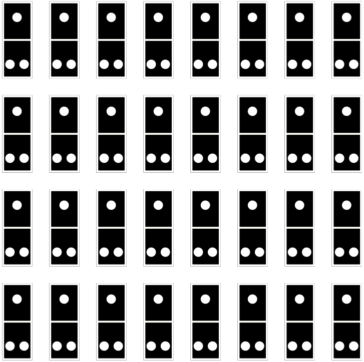


Invitation: How many ways to tile a chessboard?

I give you a standard **8×8 chessboard...**

...and 32 standard **2×1 dominoes...**

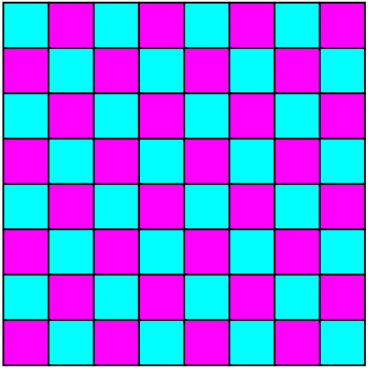


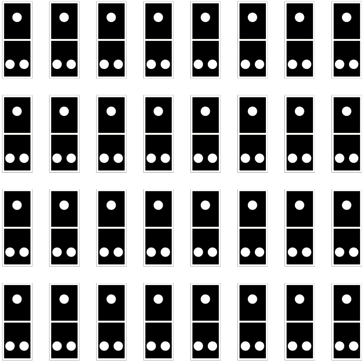


Invitation: How many ways to tile a chessboard?

I give you a standard **8×8 chessboard...**

...and 32 standard **2×1 dominoes...**



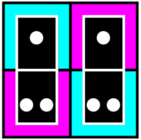


...and I ask: **How many ways can you cover the chessboard with the dominoes?**

Counting Domino Tilings

A four-step process

1. Reformulate the **tiling problem** as a **graph matching** problem.
2. Write down a matrix whose **permanent** counts graph matchings.
3. Convert this permanent into a **determinant**.
4. **Evaluate** this determinant.



Counting Domino Tilings

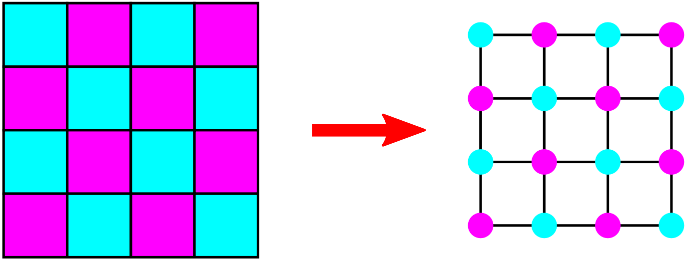
A four-step process

1. Reformulate the **tiling problem** as a **graph matching** problem.
2. Write down a matrix whose **permanent** counts graph matchings.
3. Convert this permanent into a **determinant**.
4. **Evaluate** this determinant.

Counting Domino Tilings

Step 1: Recast the problem in terms of **graph matchings.**

To the $N \times N$ chessboard we can associate a **graph** with N^2 vertices.



Graph (informal definition): Basically just a bunch of **vertices**, some pairs of which are connected by **edges**.

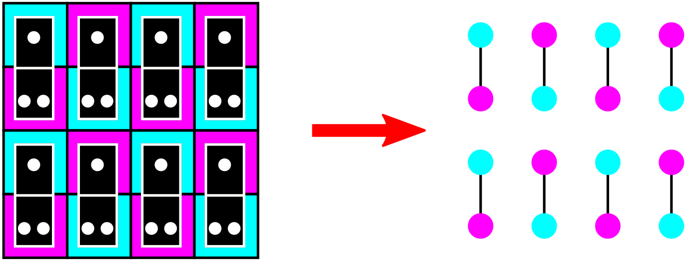
Graph (more formal definition): A *graph* is a triple (V, E, ϕ) where V is a set (whose elements are called *vertices*), E is a set (whose elements are called *edges*), and $\phi : E \rightarrow V \times V$ is a map associating each element of E with a pair of elements in V.

Counting Domino Tilings

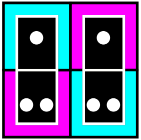
Step 1: Recast the problem in terms of **graph matchings.**

Each tiling of the board corresponds to a **perfect matching of the graph**.

Perfect matching: A subset of the edges such that each vertex is connected to precisely 1 edge.



$\Longrightarrow$ **Counting tilings is equivalent to counting perfect matchings.**

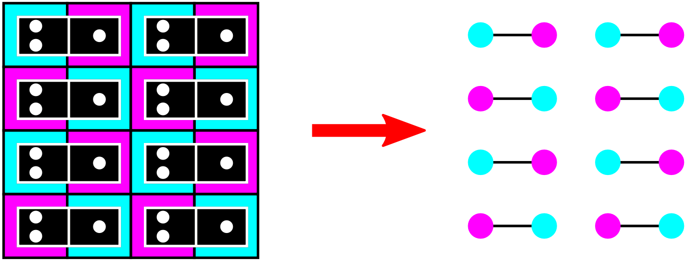


Counting Domino Tilings

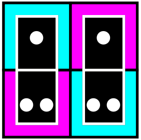
Step 1: Recast the problem in terms of **graph matchings.**

Each tiling of the board corresponds to a **perfect matching of the graph**.

Perfect matching: A subset of the edges such that each vertex is connected to precisely 1 edge.



$\Longrightarrow$ **Counting tilings is equivalent to counting perfect matchings.**

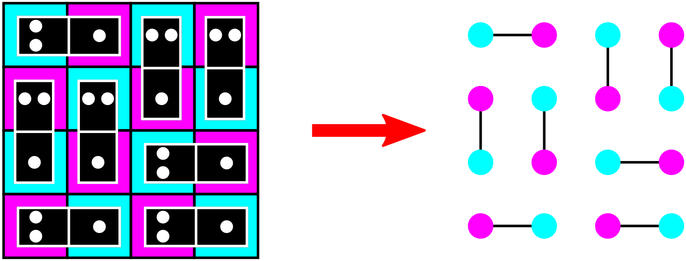


Counting Domino Tilings

Step 1: Recast the problem in terms of **graph matchings.**

Each tiling of the board corresponds to a **perfect matching of the graph**.

Perfect matching: A subset of the edges such that each vertex is connected to precisely 1 edge.



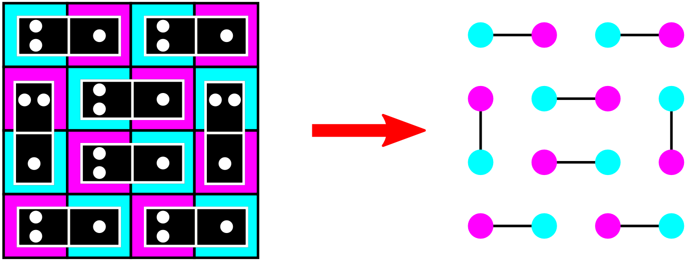
$\Longrightarrow$ **Counting tilings is equivalent to counting perfect matchings.**

Counting Domino Tilings

Step 1: Recast the problem in terms of **graph matchings.**

Each tiling of the board corresponds to a **perfect matching of the graph**.

Perfect matching: A subset of the edges such that each vertex is connected to precisely 1 edge.



$\Longrightarrow$ **Counting tilings is equivalent to counting perfect matchings.**

Counting Domino Tilings

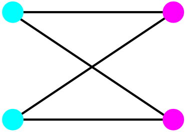
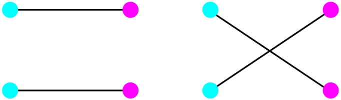
A four-step process

1. Reformulate the **tiling problem** as a **graph matching** problem.
2. Write down a matrix whose **permanent** counts graph matchings.
3. Convert this permanent into a **determinant**.
4. **Evaluate** this determinant.

Counting Domino Tilings

Practice counting perfect matchings in **bipartite** graphs

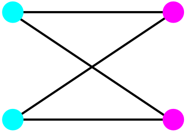
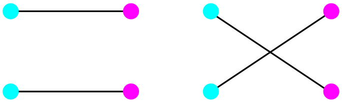
Bipartite graph: A graph whose vertices can be partitioned into two subsets ("colors") T and P such that all graph edges run from T to P, never within T or within P.

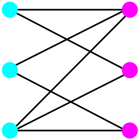
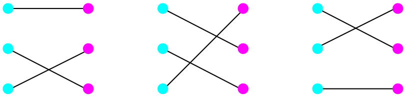
This graph...	...has **2** perfect matchings.
	

Counting Domino Tilings

Practice counting perfect matchings in **bipartite** graphs

Bipartite graph: A graph whose vertices can be partitioned into two subsets ("colors") T and P such that all graph edges run from T to P, never within T or within P.

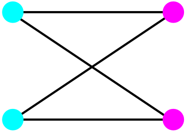
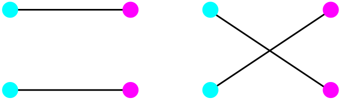
This graph...	...has **2** perfect matchings.
	

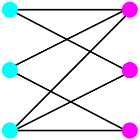
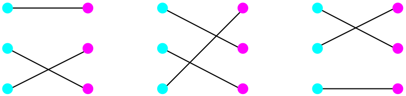
This graph...	...has **3** perfect matchings.
	

Counting Domino Tilings

Practice counting perfect matchings in **bipartite** graphs

Bipartite graph: A graph whose vertices can be partitioned into two subsets ("colors") T and P such that all graph edges run from T to P, never within T or within P.

This graph...	...has **2** perfect matchings.
	

This graph...	...has **3** perfect matchings.
	

For small graphs, we can count perfect matchings by hand... **what about larger graphs?**

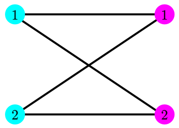
Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

The **bipartite adjacency matrix** will help us count graph matchings.

For a bipartite graph with T turquoise vertices and P purple vertices, the **bipartite adjacency matrix** $\mathbf{A}$ is a $T \times P$ matrix with entries

$$A_{tp} = \begin{cases} 1, & \text{if turquoise vertex } t \text{ is connected to purple vertex } p \\ 0, & \text{otherwise} \end{cases}$$

Graph	Bipartite adjacency matrix
	$\begin{array}{cc} & \begin{matrix} \textcircled{1} & \textcircled{2} \end{matrix} \\ \begin{matrix} \textcircled{1} \\ \textcircled{2} \end{matrix} & \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{array}$

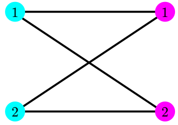
Counting Domino Tilings

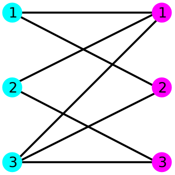
Step 2: Count the number of perfect graph matchings.

The **bipartite adjacency matrix** will help us count graph matchings.

For a bipartite graph with T turquoise vertices and P purple vertices, the **bipartite adjacency matrix** $\mathbf{A}$ is a $T \times P$ matrix with entries

$$A_{tp} = \begin{cases} 1, & \text{if turquoise vertex } t \text{ is connected to purple vertex } p \\ 0, & \text{otherwise} \end{cases}$$

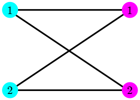
Graph	Bipartite adjacency matrix
	$\begin{array}{c c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{array}$

Graph	Bipartite adjacency matrix
	$\begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$

Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

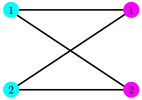
To each possible matching, we assign a numerical value computed as a **product of entries in $\mathbf{A}$**.

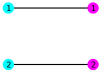
Graph	Bipartite adjacency matrix
	$\begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{c}\begin{array}{cc} 1 & 2 \end{array} \\ \left(\begin{array}{cc} 1 & 1 \\ 1 & 1 \end{array}\right)\end{array}$

Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

To each possible matching, we assign a numerical value computed as a **product of entries in $\mathbf{A}$**.

Graph	Bipartite adjacency matrix
	$\begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{array}$

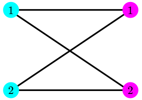


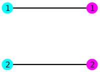
$$\begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{pmatrix} 1 & \\ & 1 \end{pmatrix} \end{array} \quad \boxed{A_{11}A_{22} = 1 \cdot 1 = 1}$$

Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

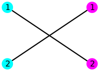
To each possible matching, we assign a numerical value computed as a **product of entries in $\mathbf{A}$**.

Graph	Bipartite adjacency matrix
	$\begin{array}{c\|cc} & 1 & 2 \\ \hline 1 & 1 & 1 \\ 2 & 1 & 1 \end{array}$



$$\begin{array}{c|cc} & 1 & 2 \\ \hline 1 & 1 & \\ 2 & & 1 \end{array}$$

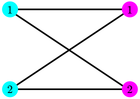
$$\boxed{A_{11}A_{22} = 1 \cdot 1 = 1}$$

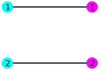


Counting Domino Tilings

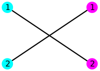
Step 2: Count the number of perfect graph matchings.

To each possible matching, we assign a numerical value computed as a **product of entries in $\mathbf{A}$**.

Graph	Bipartite adjacency matrix
	$\begin{array}{c\|cc} & 1 & 2 \\ \hline 1 & 1 & 1 \\ 2 & 1 & 1 \end{array}$



$$\begin{array}{c|cc} & 1 & 2 \\ \hline 1 & 1 & \\ 2 & & 1 \end{array} \qquad \boxed{A_{11}A_{22} = 1 \cdot 1 = 1}$$

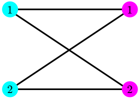


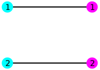
$$\begin{array}{c|cc} & 1 & 2 \\ \hline 1 & & 1 \\ 2 & 1 & \end{array} \qquad \boxed{A_{12}A_{21} = 1 \cdot 1 = 1}$$

Counting Domino Tilings

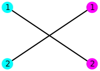
Step 2: Count the number of perfect graph matchings.

To each possible matching, we assign a numerical value computed as a **product of entries in $\mathbf{A}$**.

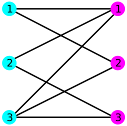
Graph	Bipartite adjacency matrix
	$\begin{array}{c c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{array}$

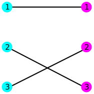


$\begin{array}{c c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{pmatrix} 1 & \\ & 1 \end{pmatrix} \end{array}$ $\boxed{A_{11}A_{22} = 1 \cdot 1 = 1}$



$\begin{array}{c c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{pmatrix} & 1 \\ 1 & \end{pmatrix} \end{array}$ $\boxed{A_{12}A_{21} = 1 \cdot 1 = 1}$

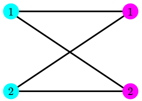
Graph	Bipartite adjacency matrix
	$\begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$

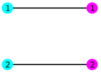


Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

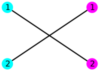
To each possible matching, we assign a numerical value computed as a **product of entries in $\mathbf{A}$**.

Graph	Bipartite adjacency matrix
	$\begin{array}{ccc} & 1 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{array}$



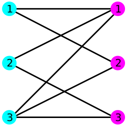
$$\begin{array}{ccc} & 1 & 2 \\ 1 & 1 & \\ 2 & & 1 \end{array}$$

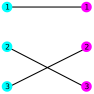
$$\boxed{A_{11}A_{22} = 1 \cdot 1 = 1}$$



$$\begin{array}{ccc} & 1 & 2 \\ 1 & & 1 \\ 2 & 1 & \end{array}$$

$$\boxed{A_{12}A_{21} = 1 \cdot 1 = 1}$$

Graph	Bipartite adjacency matrix
	$\begin{array}{cccc} & 1 & 2 & 3 \\ 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 3 & 1 & 1 & 1 \end{array}$



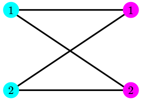
$$\begin{array}{cccc} & 1 & 2 & 3 \\ 1 & 1 & & \\ 2 & & & 1 \\ 3 & & 1 & \end{array}$$

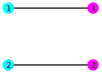
$$\boxed{A_{11}A_{23}A_{32} = 1 \cdot 1 \cdot 1 = 1}$$

Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

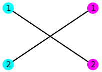
To each possible matching, we assign a numerical value computed as a **product of entries in $\mathbf{A}$**.

Graph	Bipartite adjacency matrix
	$\begin{array}{ccc} & 1 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{array}$



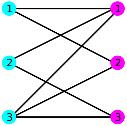
$\begin{array}{ccc} & 1 & 2 \\ 1 & 1 & \\ 2 & & 1 \end{array}$

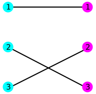
$\boxed{A_{11}A_{22} = 1 \cdot 1 = 1}$



$\begin{array}{ccc} & 1 & 2 \\ 1 & & 1 \\ 2 & 1 & \end{array}$

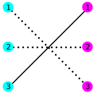
$\boxed{A_{12}A_{21} = 1 \cdot 1 = 1}$

Graph	Bipartite adjacency matrix
	$\begin{array}{cccc} & 1 & 2 & 3 \\ 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 3 & 1 & 1 & 1 \end{array}$



$\begin{array}{cccc} & 1 & 2 & 3 \\ 1 & 1 & & \\ 2 & & & 1 \\ 3 & & 1 & \end{array}$

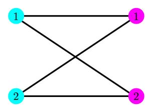
$\boxed{A_{11}A_{23}A_{32} = 1 \cdot 1 \cdot 1 = 1}$



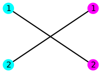
Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

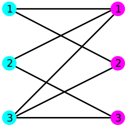
To each possible matching, we assign a numerical value computed as a **product of entries in $\mathbf{A}$**.

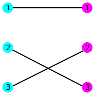
Graph	Bipartite adjacency matrix
	$\begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{array}$

$\begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{pmatrix} 1 & \\ & 1 \end{pmatrix} \end{array}$ $\boxed{A_{11}A_{22} = 1 \cdot 1 = 1}$

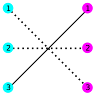


$\begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{pmatrix} & 1 \\ 1 & \end{pmatrix} \end{array}$ $\boxed{A_{12}A_{21} = 1 \cdot 1 = 1}$

Graph	Bipartite adjacency matrix
	$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$



$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & & \\ & & 1 \\ & 1 & \end{pmatrix} \end{array}$ $\boxed{A_{11}A_{23}A_{32} = 1 \cdot 1 \cdot 1 = 1}$



$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} & & 0 \\ & 0 & \\ 1 & & \end{pmatrix} \end{array}$ $\boxed{A_{13}A_{22}A_{31} = 0 \cdot 0 \cdot 1 = 0}$

Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

For a graph with N vertices (of each color), let σ be a **permutation** on $1 \cdots N$, and consider the product $A_{1\sigma(1)}A_{2\sigma(2)}\cdots A_{N\sigma(N)}$.

We have $A_{1\sigma(1)}A_{2\sigma(2)}\cdots A_{N\sigma(N)} = \begin{cases} \mathbf{1}, & \text{if the graph has a matching with } n \leftrightarrow \sigma(n)\ \forall n \\ \mathbf{0}, & \text{if there is no such matching} \end{cases}$

Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

For a graph with N vertices (of each color), let σ be a **permutation** on $1 \cdots N$, and consider the product $A_{1\sigma(1)}A_{2\sigma(2)}\cdots A_{N\sigma(N)}$.

We have $A_{1\sigma(1)}A_{2\sigma(2)}\cdots A_{N\sigma(N)} = \begin{cases} \mathbf{1}, & \text{if the graph has a matching with } n \leftrightarrow \sigma(n)\ \forall n \\ \mathbf{0}, & \text{if there is no such matching} \end{cases}$

$\Longrightarrow$ count perfect matchings by **summing over all permutations:**

$$\boxed{\sum_{\sigma} A_{1\sigma(1)}A_{2\sigma(2)}\cdots A_{N\sigma(N)} = \textbf{number of perfect matchings.}}$$

Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

For a graph with N vertices (of each color), let σ be a **permutation** on $1 \cdots N$, and consider the product $A_{1\sigma(1)}A_{2\sigma(2)}\cdots A_{N\sigma(N)}$.

We have $A_{1\sigma(1)}A_{2\sigma(2)}\cdots A_{N\sigma(N)} = \begin{cases} \mathbf{1}, & \text{if the graph has a matching with } n \leftrightarrow \sigma(n)\ \forall n \\ \mathbf{0}, & \text{if there is no such matching} \end{cases}$

$\Longrightarrow$ count perfect matchings by **summing over all permutations:**

$$\boxed{\sum_{\sigma} A_{1\sigma(1)}A_{2\sigma(2)}\cdots A_{N\sigma(N)} = \textbf{number of perfect matchings.}}$$

Where have you seen this sum before?

Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

Graph matchings are counted by the **permanent** of the bipartite adjacency matrix.

$$\boxed{\text{perm } \mathbf{A} \equiv \sum_{\sigma} A_{1\sigma(1)} A_{2\sigma(2)} \cdots A_{N\sigma(N)}}$$

Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

Graph matchings are counted by the **permanent** of the bipartite adjacency matrix.

$$\boxed{\text{perm } \mathbf{A} \equiv \sum_{\sigma} A_{1\sigma(1)} A_{2\sigma(2)} \cdots A_{N\sigma(N)}}$$

This formula is tantalizingly similar to the definition of the **determinant**...

$$\boxed{\det \mathbf{A} \equiv \sum_{\sigma} (\text{sign } \sigma) A_{1\sigma(1)} A_{2\sigma(2)} \cdots A_{N\sigma(N)}}$$

Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

Graph matchings are counted by the **permanent** of the bipartite adjacency matrix.

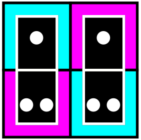
$$\boxed{\text{perm } \mathbf{A} \equiv \sum_{\sigma} A_{1\sigma(1)} A_{2\sigma(2)} \cdots A_{N\sigma(N)}}$$

This formula is tantalizingly similar to the definition of the **determinant**...

$$\boxed{\text{det } \mathbf{A} \equiv \sum_{\sigma} (\text{sign } \sigma) A_{1\sigma(1)} A_{2\sigma(2)} \cdots A_{N\sigma(N)}}$$

But the similarities are deceptive!

- **det $\mathbf{A}$** can be **computed efficiently** (in time polynomial in dim $\mathbf{A}$)...
- but in general there is **no efficient algorithm** for **perm $\mathbf{A}$**. (Valiant, 1979).



Counting Domino Tilings

Step 2: Count the number of perfect graph matchings.

Graph matchings are counted by the **permanent** of the bipartite adjacency matrix.

$$\boxed{\text{perm } \mathbf{A} \equiv \sum_{\sigma} A_{1\sigma(1)} A_{2\sigma(2)} \cdots A_{N\sigma(N)}}$$

This formula is tantalizingly similar to the definition of the **determinant**...

$$\boxed{\det \mathbf{A} \equiv \sum_{\sigma} (\text{sign } \sigma) A_{1\sigma(1)} A_{2\sigma(2)} \cdots A_{N\sigma(N)}}$$

But the similarities are deceptive!

- **det $\mathbf{A}$** can be **computed efficiently** (in time polynomial in dim $\mathbf{A}$)...
- but in general there is **no efficient algorithm** for **perm $\mathbf{A}$**. (Valiant, 1979).

Sure would be nice if we could **convert our permanent into a determinant...**

Counting Domino Tilings

A four-step process

1. Reformulate the **tiling problem** as a **graph matching** problem.
2. Write down a matrix whose **permanent** counts graph matchings.
3. Convert this permanent into a **determinant**.
4. **Evaluate** this determinant.

Counting Domino Tilings

Step 3: Convert the permanent to a determinant

Consider the various terms in the permanent and determinant expansions:

$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & \\ & 1 \end{pmatrix}$	$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$
perm $\mathbf{A}$	$+1 \cdot 1$	$+1 \cdot 1$
det $\mathbf{A}$	$+1 \cdot 1$	$-1 \cdot 1$

Counting Domino Tilings

Step 3: Convert the permanent to a determinant

Consider the various terms in the permanent and determinant expansions:

If we flip the sign of this entry, we **cancel** the minus sign in the determinant expansion!

$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ \enclose{circle}{1} & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & \\ & 1 \end{pmatrix}$	$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$
perm $\mathbf{A}$	$+1 \cdot 1$	$+1 \cdot 1$
det $\mathbf{A}$	$+1 \cdot 1$	$-1 \cdot 1$

Counting Domino Tilings

Step 3: Convert the permanent to a determinant

Consider the various terms in the permanent and determinant expansions:

If we flip the sign of this entry, we **cancel** the minus sign in the determinant expansion!

$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & \\ & 1 \end{pmatrix}$	$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$
perm $\mathbf{A}$	$+1 \cdot 1$	$+1 \cdot 1$
det $\mathbf{A}$	$+1 \cdot 1$	$-1 \cdot 1$

$\Longrightarrow$ Define a **new matrix $\tilde{\mathbf{A}}$** whose **determinant** equals the **permanent** of $\mathbf{A}$.

$\tilde{\mathbf{A}} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & \\ & 1 \end{pmatrix}$	$\begin{pmatrix} & 1 \\ -1 & \end{pmatrix}$
det $\tilde{\mathbf{A}}$	$+1 \cdot 1$	$-1 \cdot (-1)$

Counting Domino Tilings

Step 3: Convert the permanent to a determinant

Consider the various terms in the permanent and determinant expansions:

If we flip the sign of this entry, we **cancel** the minus sign in the determinant expansion!

$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & \\ & 1 \end{pmatrix}$	$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$
perm $\mathbf{A}$	$+1 \cdot 1$	$+1 \cdot 1$
det $\mathbf{A}$	$+1 \cdot 1$	$-1 \cdot 1$

$\Longrightarrow$ Define a **new matrix $\tilde{\mathbf{A}}$** whose **determinant** equals the **permanent** of $\mathbf{A}$.

$\tilde{\mathbf{A}} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & \\ & 1 \end{pmatrix}$	$\begin{pmatrix} & 1 \\ -1 & \end{pmatrix}$
det $\tilde{\mathbf{A}}$	$+1 \cdot 1$	$-1 \cdot (-1)$

det $\tilde{\mathbf{A}}$
$= 1 \cdot 1 + 1 \cdot 1$
$=$ perm $\mathbf{A}$!

Counting Domino Tilings

Step 3: Convert the permanent to a determinant

Can we do the same thing for the 3×3 matrix we considered earlier?

$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & & \\ & & 1 \\ & 1 & \end{pmatrix}$	$\begin{pmatrix} & 1 & \\ 1 & & \\ & & 1 \end{pmatrix}$	$\begin{pmatrix} & 1 & \\ & & 1 \\ 1 & & \end{pmatrix}$
perm $\mathbf{A}$	$+1 \cdot 1 \cdot 1$	$+1 \cdot 1 \cdot 1$	$+1 \cdot 1 \cdot 1$
det $\mathbf{A}$	$-1 \cdot 1 \cdot 1$	$-1 \cdot 1 \cdot 1$	$+1 \cdot 1 \cdot 1$

Counting Domino Tilings

Step 3: Convert the permanent to a determinant

Can we do the same thing for the 3×3 matrix we considered earlier?

$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & & \\ & & 1 \\ & 1 & \end{pmatrix}$	$\begin{pmatrix} & 1 & \\ 1 & & \\ & & 1 \end{pmatrix}$	$\begin{pmatrix} & 1 & \\ & & 1 \\ 1 & & \end{pmatrix}$
perm $\mathbf{A}$	$+1 \cdot 1 \cdot 1$	$+1 \cdot 1 \cdot 1$	$+1 \cdot 1 \cdot 1$
det $\mathbf{A}$	$-1 \cdot 1 \cdot 1$	$-1 \cdot 1 \cdot 1$	$+1 \cdot 1 \cdot 1$

$\tilde{\mathbf{A}} = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 1 & -1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & & \\ & & 1 \\ & -1 & \end{pmatrix}$	$\begin{pmatrix} & 1 & \\ -1 & & \\ & & 1 \end{pmatrix}$	$\begin{pmatrix} & 1 & \\ & & 1 \\ 1 & & \end{pmatrix}$
det $\mathbf{A}$	$-1 \cdot 1 \cdot (-1)$	$-1 \cdot (-1) \cdot 1$	$+1 \cdot 1 \cdot 1$

Yes! Once again we have found $\tilde{\mathbf{A}}$ such that **det $\tilde{\mathbf{A}}$ = perm $\mathbf{A}$**.

Counting Domino Tilings

Step 3: Convert the permanent to a determinant

Q. How do we intepret negative entries in $\tilde{\mathbf{A}}$ in the graph theory picture?

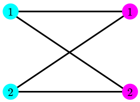
Counting Domino Tilings

Step 3: Convert the permanent to a determinant

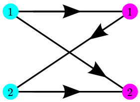
Q. How do we intepret negative entries in $\tilde{\mathbf{A}}$ in the graph theory picture?

A. Orient the graph.

- Edges running turquoise → purple are assigned $+\mathbf{1}$.
- Edges running turquoise ← purple are assigned $-\mathbf{1}$.

Graph	Bipartite adjacency matrix
	$\begin{array}{c c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{array}$

$\Rightarrow$

Oriented graph	**Oriented** BAM
	$\begin{array}{c c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \end{array}$

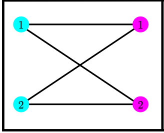
Counting Domino Tilings

Step 3: Convert the permanent to a determinant

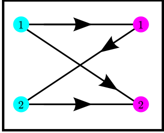
Q. How do we intepret negative entries in $\tilde{\mathbf{A}}$ in the graph theory picture?

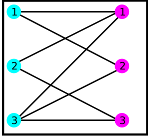
A. Orient the graph.

- Edges running turquoise → purple are assigned $+\mathbf{1}$.
- Edges running turquoise ← purple are assigned $-\mathbf{1}$.

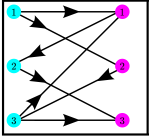
Graph	Bipartite adjacency matrix
	$\begin{array}{ccc} & 1 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{array}$

$\Rightarrow$

Oriented graph	**Oriented** BAM
	$\begin{array}{ccc} & 1 & 2 \\ 1 & 1 & 1 \\ 2 & -1 & 1 \end{array}$

Graph	Bipartite adjacency matrix
	$\begin{array}{cccc} & 1 & 2 & 3 \\ 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 3 & 1 & 1 & 1 \end{array}$

$\Rightarrow$

Oriented graph	**Oriented** BAM
	$\begin{array}{cccc} & 1 & 2 & 3 \\ 1 & 1 & 1 & 0 \\ 2 & -1 & 0 & 1 \\ 3 & 1 & -1 & 1 \end{array}$

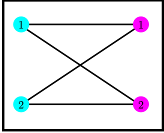
Counting Domino Tilings

Step 3: Convert the permanent to a determinant

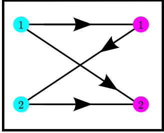
Q. How do we intepret negative entries in $\tilde{\mathbf{A}}$ in the graph theory picture?

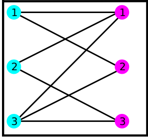
A. Orient the graph.

- Edges running turquoise → purple are assigned $+\mathbf{1}$.
- Edges running turquoise ← purple are assigned $-\mathbf{1}$.

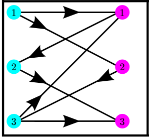
Graph	Bipartite adjacency matrix
	$\begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{array}$

$\Rightarrow$

Oriented graph	**Oriented** BAM
	$\begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \end{array}$

Graph	Bipartite adjacency matrix
	$\begin{array}{cc} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$

$\Rightarrow$

Oriented graph	**Oriented** BAM
	$\begin{array}{cc} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{pmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 1 & -1 & 1 \end{pmatrix} \end{array}$

Q. Can we **always** orient the graph in such a way that $\det \tilde{\mathbf{A}} = \operatorname{perm} \mathbf{A}$?

Counting Domino Tilings

Does the permanent → determinant trick work for *all* graphs?

Definition: Given an unoriented graph with adjacency matrix $\mathbf{A}$, a **Pfaffian orientation** is an orientation of the graph whose oriented adjacency matrix $\tilde{\mathbf{A}}$ satisfies $\textbf{det}\ \tilde{\mathbf{A}} = \textbf{perm}\ \mathbf{A}$.

Q: Does every graph have a Pfaffian orientation?

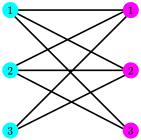
Counting Domino Tilings

Does the permanent → determinant trick work for *all* graphs?

Definition: Given an unoriented graph with adjacency matrix $\mathbf{A}$, a **Pfaffian orientation** is an orientation of the graph whose oriented adjacency matrix $\tilde{\mathbf{A}}$ satisfies $\textbf{det}\ \tilde{\mathbf{A}} = \textbf{perm}\ \mathbf{A}$.

Q: Does every graph have a Pfaffian orientation?

A: **No.** For example, this graph has no Pfaffian orientation:

Graph	BAM
	$\begin{array}{c\|ccc} & ① & ② & ③ \\ \hline ① & 1 & 1 & 1 \\ ② & 1 & 1 & 1 \\ ③ & 1 & 1 & 1 \end{array}$

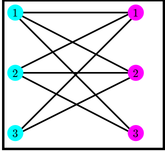
Counting Domino Tilings

Does the permanent → determinant trick work for *all* graphs?

Definition: Given an unoriented graph with adjacency matrix $\mathbf{A}$, a **Pfaffian orientation** is an orientation of the graph whose oriented adjacency matrix $\tilde{\mathbf{A}}$ satisfies **det** $\tilde{\mathbf{A}} =$ **perm** $\mathbf{A}$.

Q: Does every graph have a Pfaffian orientation?

A: **No.** For example, this graph has no Pfaffian orientation:

Graph	BAM
	$\begin{array}{c} \\ ① \\ ② \\ ③ \end{array}\begin{array}{c} \begin{array}{ccc} ① & ② & ③ \end{array} \\ \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$

However: **every <u>planar</u> graph has a Pfaffian orientation...** (Kastelyn, 1960s).

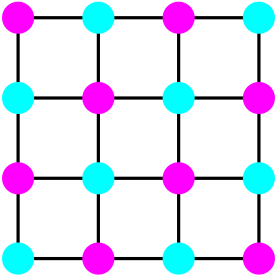
Planar graph: A graph that can be drawn on paper (i.e., in the plane) with no crossing edges.

...and, in particular, **the graph we need to count domino tilings is planar.**

Counting Domino Tilings

Pfaffian orientations of lattice graphs

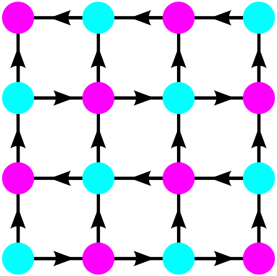
The square lattice graph (of any size)...



Counting Domino Tilings

Pfaffian orientations of lattice graphs

...has this Pfaffian orientation.



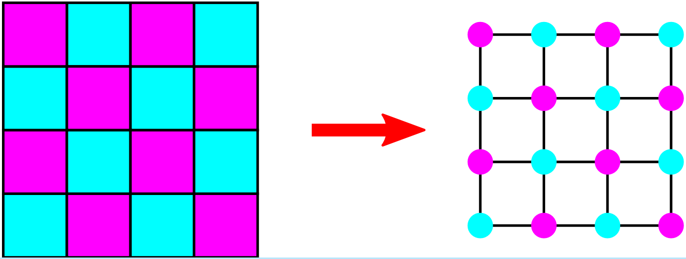
Rules for Pfaffian orientation of lattice graphs:

1. **Vertical edges** always point **upward.**
2. **Horizontal edges** point **rightward on odd rows, leftward on even rows.**

Counting Domino Tilings

Putting it all together: The solution, at last.

1. To the $M \times N$ chessboard we associate a planar bipartite graph. The number of domino tilings of the chessboard is the number of perfect matchings of the graph.



For the 4×4 case we find $\det \tilde{\mathbf{A}} = 36.$

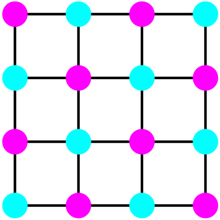
Counting Domino Tilings

Putting it all together: The solution, at last.

1. To the $M \times N$ chessboard we associate a planar bipartite graph. The number of domino tilings of the chessboard is the number of perfect matchings of the graph.
2. The number of graph matchings is the permanent of the bipartite adjacency matrix $\mathbf{A}$.

$$\#\text{ perfect matchings} = \text{perm } \mathbf{A}$$

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$



For the 4×4 case we find $\det \tilde{\mathbf{A}} = 36.$

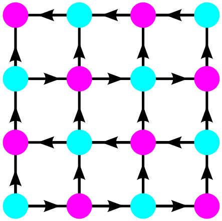
Counting Domino Tilings

Putting it all together: The solution, at last.

1. To the $M \times N$ chessboard we associate a planar bipartite graph. The number of domino tilings of the chessboard is the number of perfect matchings of the graph.
2. The number of graph matchings is the permanent of the bipartite adjacency matrix $\mathbf{A}$.
3. perm $\mathbf{A} =$ det $\tilde{\mathbf{A}}$ where $\tilde{\mathbf{A}}$ is the oriented BAM corresponding to a Pfaffian orientation.

$$\text{perm } \mathbf{A} = \det \tilde{\mathbf{A}}$$

$$\tilde{\mathbf{A}} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 1 \end{pmatrix}$$



For the 4×4 case we find $\det \tilde{\mathbf{A}} = 36.$

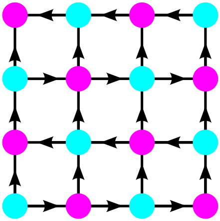
Counting Domino Tilings

Putting it all together: The solution, at last.

1. To the $M \times N$ chessboard we associate a planar bipartite graph. The number of domino tilings of the chessboard is the number of perfect matchings of the graph.
2. The number of graph matchings is the permanent of the bipartite adjacency matrix $\mathbf{A}$.
3. perm $\mathbf{A} =$ det $\tilde{\mathbf{A}}$ where $\tilde{\mathbf{A}}$ is the oriented BAM corresponding to a Pfaffian orientation.
4. **And thus: # of domino tilings** $= \det \tilde{\mathbf{A}}$.

$$\text{perm } \mathbf{A} = \det \tilde{\mathbf{A}}$$

$$\tilde{\mathbf{A}} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 1 \end{pmatrix}$$

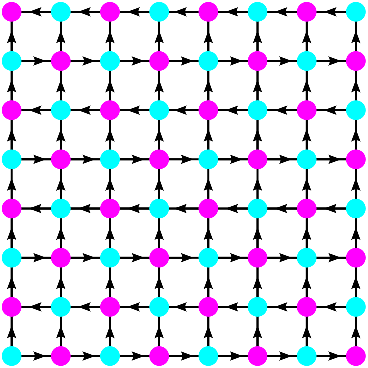


For the 4×4 case we find $\det \tilde{\mathbf{A}} = 36$.

Counting Domino Tilings

Putting it all together: The solution, at last.

Finally, for the 8x8 case...



Counting Domino Tilings

Putting it all together: The solution, at last.

Finally, for the 8x8 case...

$$\tilde{\mathbf{A}}=\begin{pmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1
\end{pmatrix}$$

Counting Domino Tilings

Putting it all together: The solution, at last.

Finally, for the 8x8 case...

$$\tilde{\mathbf{A}}=\begin{pmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1
\end{pmatrix}$$

...we obtain **det $\tilde{\mathbf{A}}$ =12,988,816.**

Counting Domino Tilings

A four-step process

1. Reformulate the **tiling problem** as a **graph matching** problem.
2. Write down a matrix whose **permanent** counts graph matchings.
3. Convert this permanent into a **determinant**.
4. **Evaluate** this determinant.

Counting Domino Tilings

Analytical Evaluation of Circulant Matrix Determinants

It turns out that **det $\tilde{\mathbf{A}}$** can be evaluated **analytically** for the chessboard tiling problem.

The key idea is to exploit the properties of **circulant matrices.**

Circulant matrix: Each row is a shifted version of the previous row, e.g.: $\begin{pmatrix} A_1 & A_2 & A_3 & A_4 & A_5 \\ A_5 & A_1 & A_2 & A_3 & A_4 \\ A_4 & A_5 & A_1 & A_2 & A_3 \\ A_3 & A_4 & A_5 & A_1 & A_2 \\ A_3 & A_4 & A_5 & A_1 & A_2 \end{pmatrix}$

Consider this 5×5 **cyclic permutation matrix:**

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Clearly the effect of multiplying any vector by this matrix is to shift every element to the previous slot (and the first element loops around to the last slot):

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix} = \begin{pmatrix} v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_1 \end{pmatrix}$$

Now put $\xi = e^{2\pi i/5}$ (a 5th root of unity) and consider this vector:

$$\mathbf{v} = \begin{pmatrix} 1 \\ \xi \\ \xi^2 \\ \xi^3 \\ \xi^4 \end{pmatrix}$$

Evidently, excuting the shift operation on this matrix is equivalent to multiplying each entry by ξ, so

$\mathbf{M}\mathbf{v} = \xi\mathbf{v}$, i.e. **$\xi$ is an eigenvalue of $\mathbf{M}$.**

Counting Domino Tilings

Analytical Evaluation of Circulant Matrix Determinants

Extending the logic of the previous slide, it's easy to write down analytical formulas for the **eigenvalues of an arbitrary circulant matrix:**

The eigenvalues of

$$\mathbf{M} = \begin{pmatrix} A_1 & A_2 & A_3 & A_4 & A_5 \\ A_5 & A_1 & A_2 & A_3 & A_4 \\ A_4 & A_5 & A_1 & A_2 & A_3 \\ A_3 & A_4 & A_5 & A_1 & A_2 \\ A_3 & A_4 & A_5 & A_1 & A_2 \end{pmatrix}$$

are ($\xi = e^{2\pi i/5}$)

$$\boxed{A_1 + A_2\xi^n + A_3\xi^{2n} + A_4\xi^{3n} + A_5\xi^{5n}, \quad n = 1 \cdots 5}$$

The **determinant** is just the product of the eigenvalues:

$$\boxed{\det \mathbf{M} = \prod_{n=1}^{5}\Big(A_1 + A_2\xi^n + A_3\xi^{2n} + A_4\xi^{3n} + A_5\xi^{5n}\Big).}$$

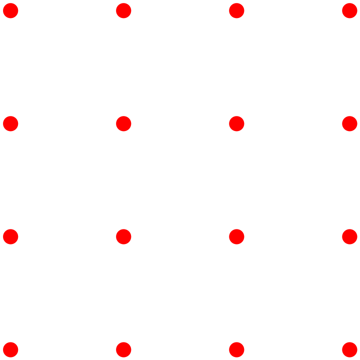
By playing these games on the domino-tiling matrix $\tilde{\mathbf{A}}$, we find an astonishing **analytical formula:**

$$\boxed{\texttt{Tilings}(M,N) = \det \tilde{\mathbf{A}}(M,N) = 4^{\frac{MN}{4}} \cdot \prod_{k=1}^{\frac{M}{2}} \prod_{l=1}^{\frac{N}{2}} \left(\cos^2\frac{k\pi}{M+1} + \cos^2\frac{l\pi}{N+1}\right).}$$

The Partition Function of the 2D Ising Model

Introduction to the 2D Ising Ferromagnet

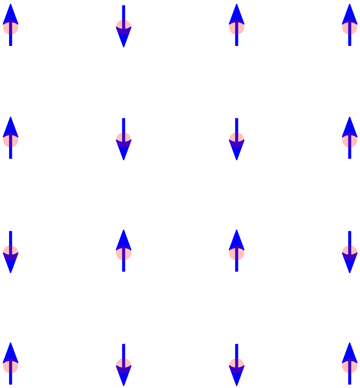
- Begin with an $N \times N$ square grid.



The Partition Function of the 2D Ising Model

Introduction to the 2D Ising Ferromagnet

- Begin with an $N \times N$ square grid.
- Put a microscopic magnet (a **spin**) on each site. Each spin may point up or down.



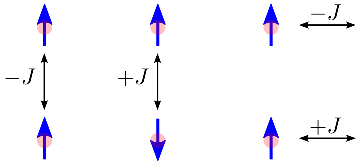
The Partition Function of the 2D Ising Model

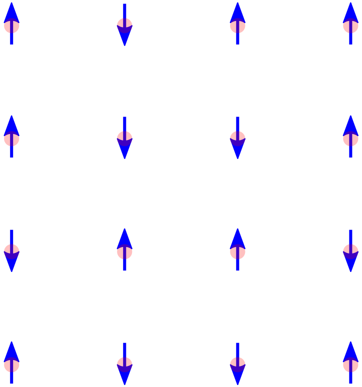
Introduction to the 2D Ising Ferromagnet

- Begin with an $N \times N$ square grid.

- Put a microscopic magnet (a **spin**) on each site. Each spin may point up or down.

- Assign to each adjacent pair of spins an **interaction energy:** $-J$ if both up or both down, $+J$ otherwise.





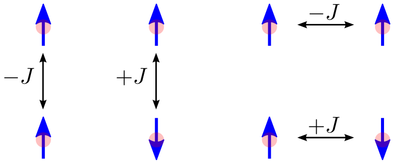
The Partition Function of the 2D Ising Model

Introduction to the 2D Ising Ferromagnet

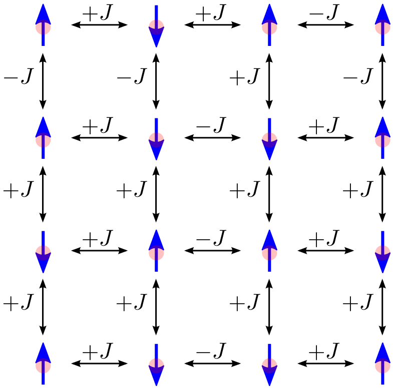
- Begin with an $N \times N$ square grid.

- Put a microscopic magnet (a **spin**) on each site. Each spin may point up or down.

- Assign to each adjacent pair of spins an **interaction energy:** $-J$ if both up or both down, $+J$ otherwise.



- Sum all interaction energies to **assign an energy to the entire configuration.**



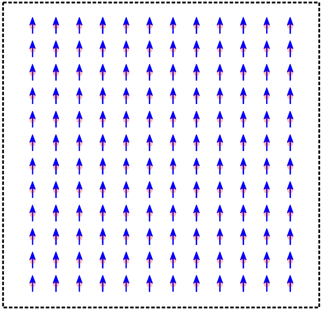
$$\boxed{\text{Total Energy: } E = +10J}$$

The Partition Function of the 2D Ising Model

Free energy: balance between

Q: In what sort of configuration would we **expect** to find an Ising ferromagnet?

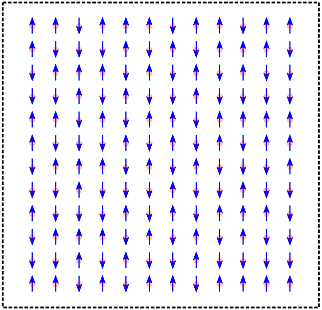
An **ordered** state...



We arrive at this kind of configuration by **minimizing energy:**

minimize E

or a **disordered** state?



We arrive at this kind of configuration by **maximizing entropy:**

maximize S

The Partition Function of the 2D Ising Model

Free energy: a balance between minimizing energy and maximizing entropy

- If energy differences are significant, we find the expected configuration by **minimizing energy**, i.e. by solving an optimization problem of the form

$$\boxed{\textbf{minimize } E}$$

- If energy differences are not significant, we find the expected configuration by **maximizing entropy**, i.e. by solving an problem of the form

$$\boxed{\textbf{maximize } S}$$

Q: If energy differences are significant ... **relative to what??**

The Partition Function of the 2D Ising Model

Free energy: a balance between minimizing energy and maximizing entropy

- If energy differences are significant, we find the expected configuration by **minimizing energy**, i.e. by solving an optimization problem of the form

$$\boxed{\textbf{minimize } E}$$

- If energy differences are not significant, we find the expected configuration by **maximizing entropy**, i.e. by solving an problem of the form

$$\boxed{\textbf{maximize } S}$$

Q: If energy differences are significant ... **relative to what??**

Answer: **Temperature.**

- At **low** T, energy differences between configurations are significant, and the expected configuration is obtained by solving $\boxed{\textbf{minimize } E.}$
- At **high** T, energy differences between configurations are insignificant, and the expected configuration is obtained by solving $\boxed{\textbf{maximize } S.}$

The quantity which interpolates between these two imperatives is the **free energy** $F = E - TS$**:**

$$\boxed{\text{minimize } E} \quad \xleftarrow[\text{low } T]{} \quad \boxed{\text{minimize } F = E - TS} \quad \xrightarrow[\text{high } T]{} \quad \boxed{\text{maximize } S}$$

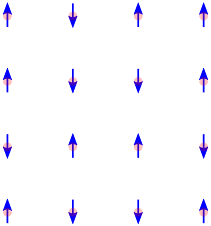
The Partition Function of the 2D Ising Model

Conversion to a graph matching problem

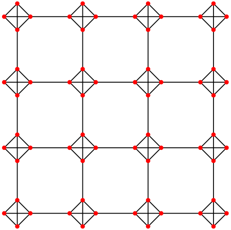
The free energy F is related to the **partition function** Z :

$$F = -T \log Z, \qquad \boxed{Z \equiv \sum_{\text{configurations}} e^{-E/T}}$$

Z for this Ising lattice...



...can be computed by
counting matchings in this graph.



References:

- Hurst and Green, *Journal of Chemical Physics*, **33** 1069 (1960)
- Kastelyn, *Journal of Mathematical Physics*, **4** 287 (1963)

The Partition Function of the 2D Ising Model

The Result

The result for the partition function of an $N \times N$ Ising ferromagnet is ($x \equiv \tanh(J/T)$)

$$\boxed{Z_N^2(T) = \prod_{p=1}^{N}\prod_{q=1}^{N}\left\{(1+x^2)^2 - 2x(1-x^2)\left[\cos\frac{2\pi p}{N} + \cos\frac{2\pi q}{N}\right]\right\}}$$

Determinants That Count

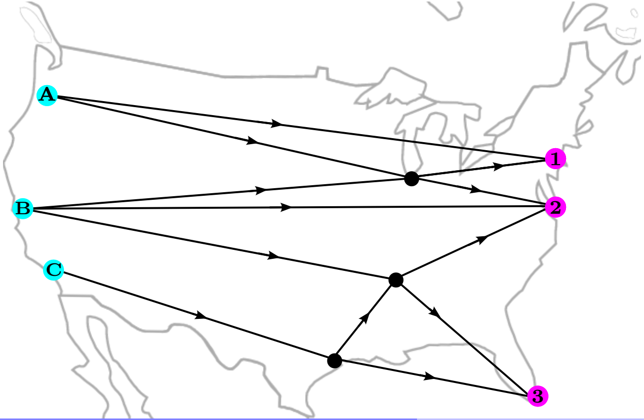
An Outline of this Talk

1. Determinants that count **domino tilings**:
   The Kastelyn-Percus-Hurst-Green Method
2. Determinants that count **lattice path systems**:
   The Gessel-Viennot Method

Invitation: How many ways to crisscross the country?

Counting vertex-disjoint lattice path systems

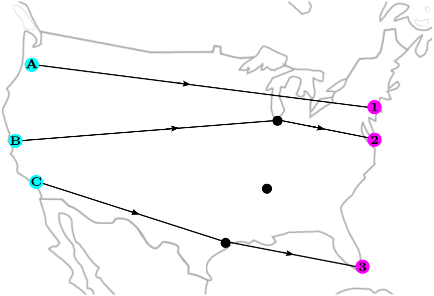
3 traveling salespeople start in 3 origin cities (A,B,C). How many different ways can they travel to 3 destination cities (1,2,3) **without overlapping at any city?**



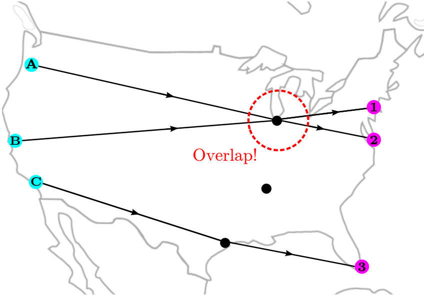
Invitation: How many ways to crisscross the country?

Counting vertex-disjoint lattice path systems

Non-overlapping (**"vertex-disjoint"**) vs. overlapping path systems:



OK (Vertex disjoint)

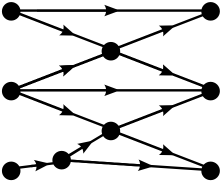


No good! **Not vertex disjoint.**

Counting Vertex-Disjoint Path Systems

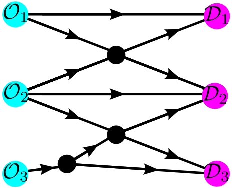
The Method of Gessel and Viennot

1. Start with a **directed, acyclic** graph. (*Acyclic*: no nontrivial paths from any vertex back to itself.)



Counting Vertex-Disjoint Path Systems

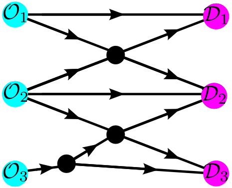
The Method of Gessel and Viennot



1. Start with a **directed, acyclic** graph. (*Acyclic*: no nontrivial paths from any vertex back to itself.)
2. Identify a set of N **origin** vertices $\{\mathcal{O}_n\}$ and a set of N **destination** vertices $\{\mathcal{D}_n\}$.

Counting Vertex-Disjoint Path Systems

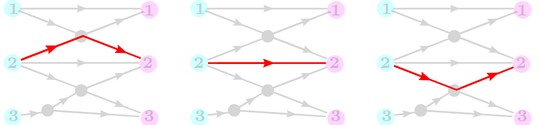
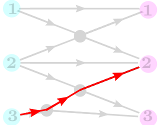
The Method of Gessel and Viennot



1. Start with a **directed, acyclic** graph. (*Acyclic*: no nontrivial paths from any vertex back to itself.)
2. Identify a set of N **origin** vertices $\{\mathcal{O}_n\}$ and a set of N **destination** vertices $\{\mathcal{D}_n\}$.
3. Form the $N \times N$ **path matrix** $\mathbf{P}$, where

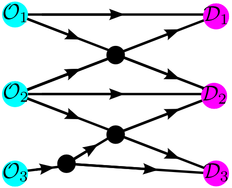
$$\boxed{P_{ij} = \# \text{ paths from vertex } \mathcal{O}_i \text{ to vertex } \mathcal{D}_j.}$$

For example:

$P_{22} = 3$	$P_{32} = 1$
	

Counting Vertex-Disjoint Path Systems

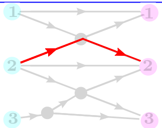
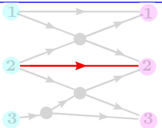
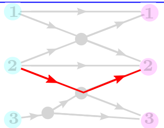
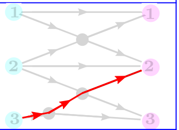
The Method of Gessel and Viennot



1. Start with a **directed, acyclic** graph. (*Acyclic*: no nontrivial paths from any vertex back to itself.)
2. Identify a set of N **origin** vertices $\{\mathcal{O}_n\}$ and a set of N **destination** vertices $\{\mathcal{D}_n\}$.
3. Form the $N \times N$ **path matrix** $\mathbf{P}$, where

$$\boxed{P_{ij} = \#\text{ paths from vertex } \mathcal{O}_i \text{ to vertex } \mathcal{D}_j.}$$

For example:

$P_{22} = 3$			$P_{32} = 1$
			

4. Then:

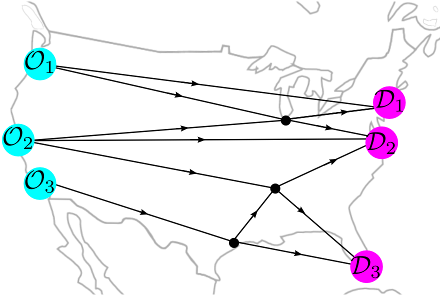
$$\boxed{\textbf{det } \mathbf{P} \textbf{=number of vertex-disjoint path systems from } \{\mathcal{O}_n\} \textbf{ to } \{\mathcal{D}_n\}.}$$

Counting Vertex-Disjoint Path Systems

Using Gessel-Viennot to solve our traveling salesman problem

For this graph with this choice of $\{\mathcal{O}_n\}$ and $\{\mathcal{D}_n\}$, the path matrix is

$$P = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

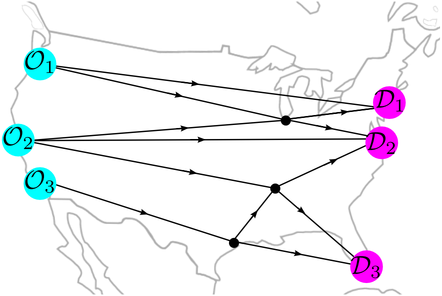


Counting Vertex-Disjoint Path Systems

Using Gessel-Viennot to solve our traveling salesman problem

For this graph with this choice of $\{\mathcal{O}_n\}$ and $\{\mathcal{D}_n\}$, the path matrix is

$$P = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$



Hence we find

 # vertex-disjoint path systems=det $\mathbf{P} = 8$

Counting Vertex-Disjoint Path Systems

Using Gessel-Viennot to solve our traveling salesman problem

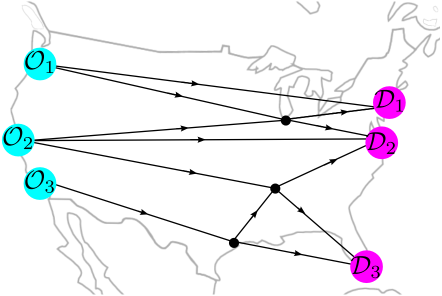
For this graph with this choice of $\{\mathcal{O}_n\}$ and $\{\mathcal{D}_n\}$, the path matrix is

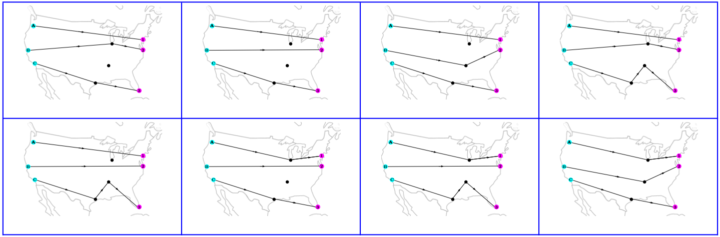
$$P = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

Hence we find

 # vertex-disjoint path systems=det $\mathrm{P} = 8$

as we found previously by hand calculation.





Counting Vertex-Disjoint Path Systems

How does Gessel-Viennot work?

Consider the various terms in the expansion of **det P:**

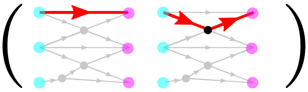
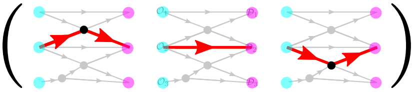
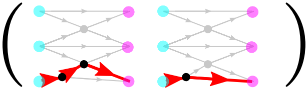
$$\det\begin{vmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{vmatrix} = 2\cdot 3\cdot 2\ -\ 2\cdot 1\cdot 1\ -\ 1\cdot 1\cdot 2$$

Counting Vertex-Disjoint Path Systems

How does Gessel-Viennot work?

Consider the various terms in the expansion of **det P:**

$$\det\begin{vmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{vmatrix} = \boxed{2\cdot 3\cdot 2} - 2\cdot 1\cdot 1 - 1\cdot 1\cdot 2$$

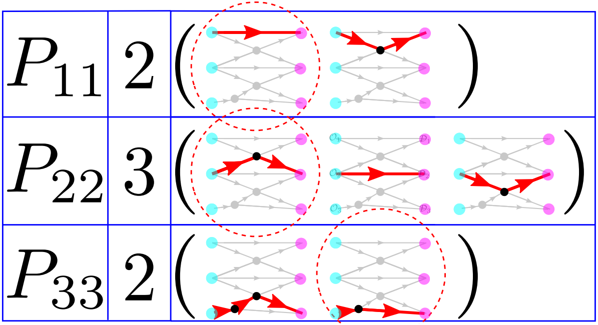
P_{11}	2	
P_{22}	3	
P_{33}	2	

Counting Vertex-Disjoint Path Systems

How does Gessel-Viennot work?

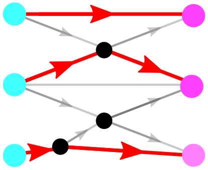
Consider the various terms in the expansion of **det P:**

$$\det \begin{vmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{vmatrix} = \boxed{2\cdot 3\cdot 2} - 2\cdot 1\cdot 1 - 1\cdot 1\cdot 2$$



P_{11}	2
P_{22}	3
P_{33}	2

This term counts both acceptable (vertex-disjoint) path systems like this:

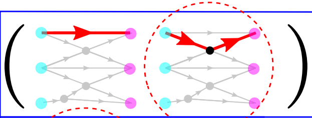
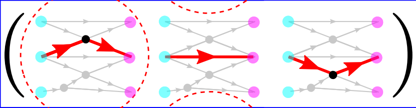
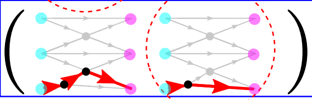


Counting Vertex-Disjoint Path Systems

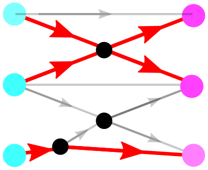
How does Gessel-Viennot work?

Consider the various terms in the expansion of **det P**:

$$\det\begin{vmatrix}2&1&0\\1&3&1\\0&1&2\end{vmatrix}=\boxed{2\cdot3\cdot2}-2\cdot1\cdot1-1\cdot1\cdot2$$

P_{11}	2	
P_{22}	3	
P_{33}	2	

...as well as unacceptable (not vertex-disjoint) path systems like this:

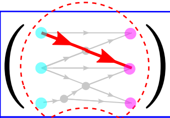
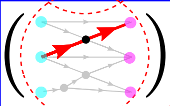
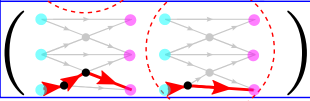


Counting Vertex-Disjoint Path Systems

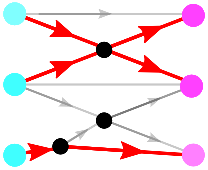
How does Gessel-Viennot work?

Consider the various terms in the expansion of **det P**:

$$\det\begin{vmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{vmatrix} = 2\cdot 3\cdot 2 \; - \; 2\cdot 1\cdot 1 \; \boxed{- \; 1\cdot 1\cdot 2}$$

P_{12}	1	
P_{21}	1	
P_{33}	2	

But the unacceptable path systems are magically cancelled by other terms in the determinant!

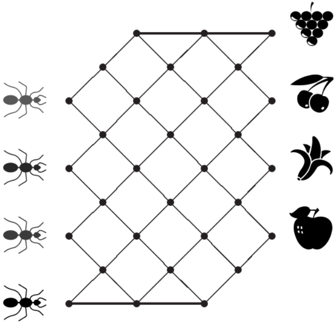


Counting Vertex-Disjoint Path Systems

Another application of Gessel-Viennot: **determined ants**

How many ways can the 4 ants propagate to the 4 fruits?

Note: All edges are directed **rightward**: either →, ↗, or ↘ .



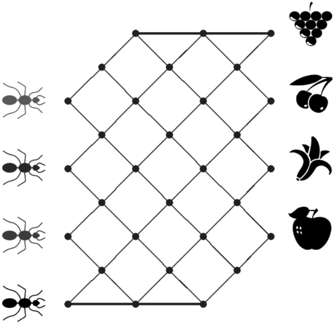
Reference: Benjamin and Cameron, *American Mathematical Monthly*, June-July 2005

Counting Vertex-Disjoint Path Systems

Another application of Gessel-Viennot: **determined ants**

How many ways can the 4 ants propagate to the 4 fruits?

Note: All edges are directed **rightward**: either →, ↗, or ↘ .



Answer:

$$\textbf{det}\begin{vmatrix} 14 & 6 & 1 & 0 \\ 20 & 15 & 6 & 1 \\ 15 & 20 & 15 & 6 \\ 6 & 15 & 20 & 14 \end{vmatrix} = \textbf{889}.$$

Reference: Benjamin and Cameron, *American Mathematical Monthly*, June-July 2005

Counting Vertex-Disjoint Path Systems

Gessel-Viennot: A technical caveat

The Gessel-Viennot method only works as we have described it for graphs and vertex sets $\{\mathcal{O}_n\}, \{\mathcal{D}_n\}$ that are **nonpermutable.**

Nonpermutable: The only vertex-disjoint path systems (VDPSs) from $\{\mathcal{O}_n\}$ to $\{\mathcal{D}_n\}$ take $\mathcal{O}_i \to \mathcal{D}_i$ for all $i = 1 \cdots N$.

More generally, in the permutable case, we have

$$\boxed{\det \mathbf{P} = \ \# \text{ of } \textbf{positive } VDPSs \ - \ \# \text{ of } \textbf{negative } VDPSs}$$

where the sign of a VDPS is the sign of the permutation that maps the origin vertices to the destination vertices, $\sigma : \{\mathcal{O}_n\} \to \{\mathcal{D}_n\}$.

Counting Vertex-Disjoint Path Systems: The Fine Print

Some more information on the method we have presented

- The method we have presented was discussed by Gessel and Viennot in 1985, although some of the ideas were anticipated earlier by Lindström.
- In addition to problems involving traveling salesmen and ants, the method can be used to enumerate a host of other combinatorial sets, including rhombic tilings, plane partitions, permutations with specified descent sets, and more.
- A thorough reference on the method presented here may be found in Chapter 5 of Aigner, *A Course In Enumeration*, Springer GTM #238, in which you can also find numerous other applications of the method.
- See also Stanley, *Enumerative Combinatorics*, Volume 1, Section 2.7.

Determinants That Count

In conclusion

1. Determinants that count **domino tilings**:
   The Kastelyn-Percus-Hurst-Green Method

2. Determinants that count **lattice path systems**:
   The Gessel-Viennot Method

It turns out that the two methods we have discussed here today are **related to each other:**

- Greg Kuperberg, "An Exploration of the Permanent-Determinant Method," *Electronic Journal of Combinatorics*, **5** R46 (1998)