How Many Ways Can We Tile a Rectangular Chessboard With Dominos?

Counting Tilings With Permanents and Determinants

Brendan W. Sullivan

Carnegie Mellon University
Undergraduate Math Club

February 20, 2013

Abstract

Consider an $m \times n$ rectangular chessboard. Suppose we want to tile this board with dominoes, where a domino is a 2×1 rectangle, and a tiling is a way to place several dominoes on the board so that all of its squares are covered but no dominos overlap or lie partially off the board. Is such a tiling possible? If so, how many are there? The first question is simple, yet the second question is quite difficult! We will answer it by reformulating the problem in terms of perfect matchings in bipartite graphs. Counting these matchings will be achieved efficiently by finding a particularly helpful matrix that describes the edges in a matching, and then finding the determinant of that matrix. Remarkably, there is even a closed-form solution!

(Note: This talk is adapted from a Chapter in Jiří Matoušek's book *Thirty-three Miniatures: Mathematical and Algorithmic Applications of Linear Algebra* [1].)

1 Introduction
- The Problem
- Explorations
- Generalizations
- Applications

2 Reformulation
- Definitions
- Matrices and Permutations
- Permanents and Determinants
- Kasteleyn Signings

3 The Main Theorem
- Graph Properties
- Theorem Statement
- Lemma 1 (and Proof)
- Lemma 2 (and Proof)
- Proof of Theorem

4 Results & Conclusions
- Summary of Method
- Applying the Method
- Closed-Form Solution
- Other Work
- References

Chessboards & Dominoes

Consider an $m \times n$ rectangular chessboard and 2×1 dominoes.

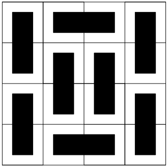
A **tiling** is a placement of dominoes that covers all the squares of the board perfectly (i.e. no overlaps, no diagonal placements, no protrusions off the board, and so on).

Chessboards & Dominoes

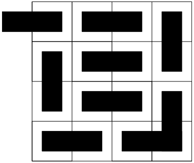
Consider an $m \times n$ rectangular chessboard and 2×1 dominoes.

A **tiling** is a placement of dominoes that covers all the squares of the board perfectly (i.e. no overlaps, no diagonal placements, no protrusions off the board, and so on).

A domino tiling of a 4×4 board



A non-tiling of a 4×4 board

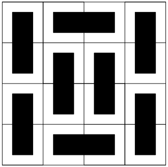


Chessboards & Dominoes

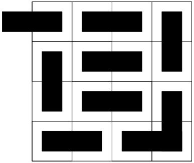
Consider an $m \times n$ rectangular chessboard and 2×1 dominoes.

A **tiling** is a placement of dominoes that covers all the squares of the board perfectly (i.e. no overlaps, no diagonal placements, no protrusions off the board, and so on).

A domino tiling of a 4×4 board



A non-tiling of a 4×4 board



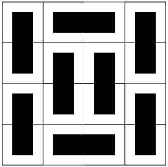
(i) For which m, n do there *exist* tilings?

Chessboards & Dominoes

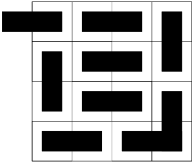
Consider an $m \times n$ rectangular chessboard and 2×1 dominoes.

A **tiling** is a placement of dominoes that covers all the squares of the board perfectly (i.e. no overlaps, no diagonal placements, no protrusions off the board, and so on).

A domino tiling of a 4×4 board



A non-tiling of a 4×4 board



(i) For which m, n do there *exist* tilings?
(ii) If there are tilings, how *many* are there?

(i) Existence of tilings: A fundamental fact

Fact: Tilings exist $\iff$ m, n are not *both* odd (i.e. mn is even)

Proof.

WOLOG m is even. Place $\frac{m}{2}$ dominoes vertically in 1st column. Repeat across n columns. □

(i) Existence of tilings: A fundamental fact

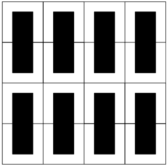
Fact: Tilings exist $\iff$ m, n are not *both* odd (i.e. mn is even)

Proof.

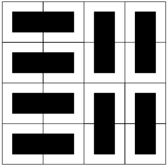
WOLOG m is even. Place $\frac{m}{2}$ dominoes vertically in 1st column. Repeat across n columns. □

Example: We will consider the 4×4 case throughout this talk.

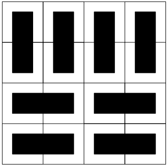
Tiling 1



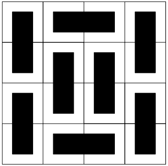
Tiling 2



Tiling 3



Tiling 4



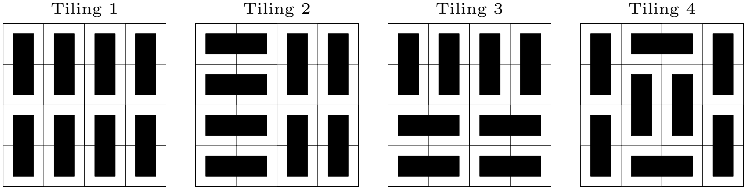
(i) Existence of tilings: A fundamental fact

Fact: Tilings exist $\iff$ m, n are not *both* odd (i.e. mn is even)

Proof.

WOLOG m is even. Place $\frac{m}{2}$ dominoes vertically in 1st column. Repeat across n columns. □

Example: We will consider the 4×4 case throughout this talk.



Note: 2 and 3 are *isomorphic*. We won't account for this. (Too hard!)

(ii) Counting tilings: A fundamental example

Consider $m = 2$. A recurrence for $T(2, n)$ is given by

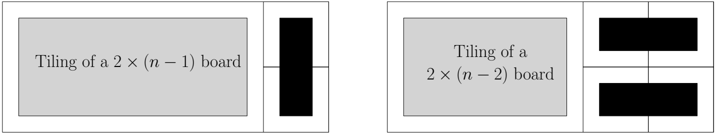
$$T(2, n) = T(2, n-1) + T(2, n-2)$$

(ii) Counting tilings: A fundamental example

Consider $m = 2$. A recurrence for $T(2, n)$ is given by

$$T(2, n) = T(2, n-1) + T(2, n-2)$$

because a tiling of a $2 \times n$ board consists of (a) a tiling of a $2 \times (n-1)$ board with a vertical domino or (b) a tiling of a $2 \times (n-2)$ board with two horizontal dominoes:

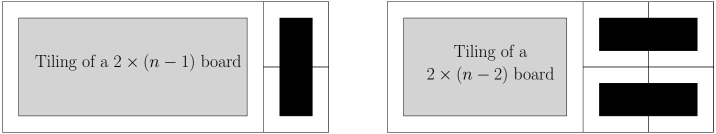


(ii) Counting tilings: A fundamental example

Consider $m = 2$. A recurrence for $T(2, n)$ is given by

$$T(2, n) = T(2, n-1) + T(2, n-2)$$

because a tiling of a $2 \times n$ board consists of (a) a tiling of a $2 \times (n-1)$ board with a vertical domino or (b) a tiling of a $2 \times (n-2)$ board with two horizontal dominoes:



Since $T(2, 1) = 1$ and $T(2, 2) = 2$ (recall: isomorphisms irrelevant) we have $T(2, n) = F_{n-1}$. It's the Fibonacci sequence!

(ii) Counting tilings: A naive recursive approach

Shouldn't we be able to adapt the $m = 2$ case to larger m?

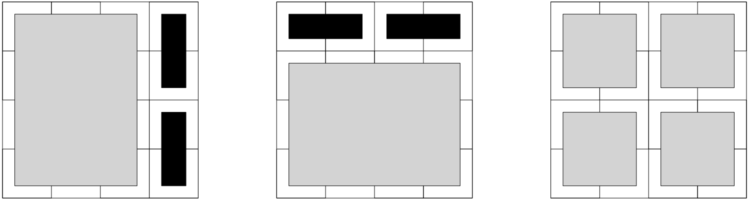
Let's try a 4×4 board.

(ii) Counting tilings: A naive recursive approach

Shouldn't we be able to adapt the $m = 2$ case to larger m?

Let's try a 4×4 board. We might write

$$T(4,4) = T(4,3) + T(3,4) + T(2,2)^4 - \ldots + \ldots$$

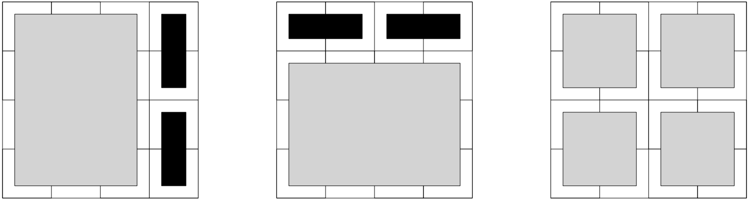


(ii) Counting tilings: A naive recursive approach

Shouldn't we be able to adapt the $m = 2$ case to larger m?

Let's try a 4×4 board. We might write

$$T(4,4) = T(4,3) + T(3,4) + T(2,2)^4 - \dots + \dots$$



This is too difficult, in general! ☹

Recursion: it's not all bad

One can prove, for example that

$$T(3,2n) = 4T(3,2n-2) - T(3,2n-4)$$

Proof.

Exercise for the reader.

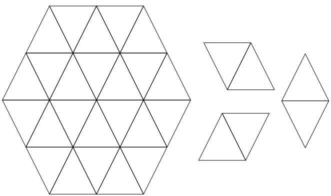
Hint: First prove

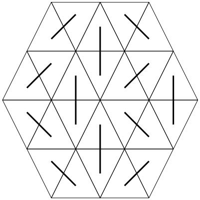
$$T(3,2n+2) = 3T(3,2n) + 2\sum_{k=0}^{n} T(3,2k)$$

□

Hexagonal Tilings

Consider a regular hexagon made of equilateral triangles, and rhombic tiles made of two such triangles.

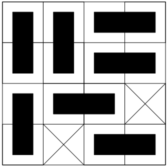




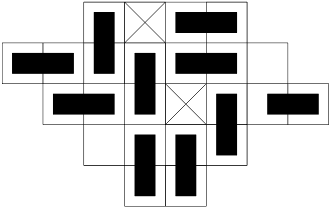
Ask the same questions of (i) existence and (ii) counting.

Altered Rectangles

What if we remove squares from the rectangular boards?



What about other crazy shapes?



Tilings, Perfect Matchings, and The Dimer Model

- **Tilings**: Popular recreational math topic. Great exercises! Tilings of the plane appear in ancient art, and reflect some deep group theoretic principles.
- **Perfect Matchings**: Useful in computer science. Algorithms for finding matchings of various forms in different types of graphs are studied for their computational complexity.
- **The Dimer Model**: Simple model used to describe thermodynamic behavior of fluids. It was the original motivation for this problem, solved in 1961 by P.W. Kasteleyn [2] and independently by Temperley & Fisher [3].

Graph Theory & Linear Algebra

We will take a seemingly roundabout approach to find $T(m, n)$. We will reformulate the problem in terms of **graphs** and then use **linear algebra** to assess properties of particular graphs. This will solve the problem!

Graph Theory & Linear Algebra

We will take a seemingly roundabout approach to find $T(m, n)$. We will reformulate the problem in terms of **graphs** and then use **linear algebra** to assess properties of particular graphs. This will solve the problem!

Fundamental idea: A domino tiling corresponds (uniquely) to a perfect matching in the underlying grid graph of the board.

Graph Theory & Linear Algebra

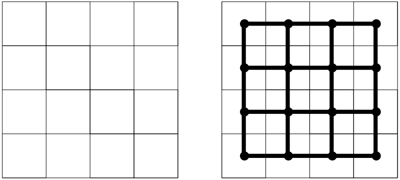
We will take a seemingly roundabout approach to find $T(m, n)$. We will reformulate the problem in terms of **graphs** and then use **linear algebra** to assess properties of particular graphs. This will solve the problem!

Fundamental idea: A domino tiling corresponds (uniquely) to a perfect matching in the underlying grid graph of the board.

Restatement: A domino tiling is characterized by which squares are covered by the same domino. We merely need to count the ways to properly assign these so that it *is* a tiling.

Example illustration

Represent the board with a dot (**vertex**) in each square and a line (**edge**) between adjacent squares (non-diagonally).

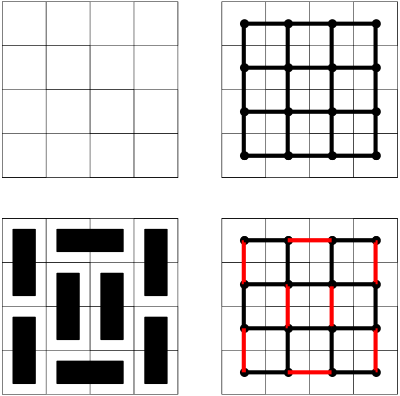


Example illustration

Represent the board with a dot (**vertex**) in each square and a line (**edge**) between adjacent squares (non-diagonally).

A tiling corresponds to a selection of these edges (and *only* these allowable edges) that *covers* every vertex.

In other terminology, this is a **perfect matching**.



Graph terminology

Definition

A **bipartite** *graph is one whose vertices can be separated into two parts, so that edges only go between parts (i.e. no internal edges in a part).*

A **perfect matching** *in a graph is a selection of edges that covers each vertex exactly once.*

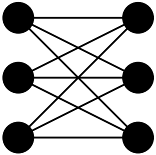
Graph terminology

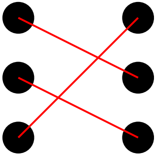
Definition

A **bipartite** *graph is one whose vertices can be separated into two parts, so that edges only go between parts (i.e. no internal edges in a part).*

A **perfect matching** *in a graph is a selection of edges that covers each vertex exactly once.*

Example: $K_{3,3}$, the *complete* bipartite graph.

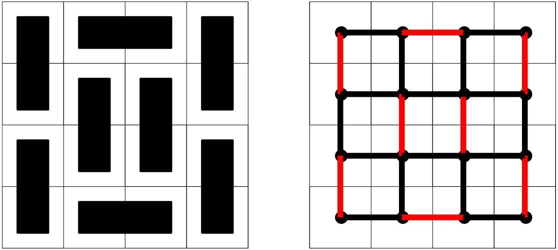




(Note: In general, a perfect matching *requires* an even number of vertices.)

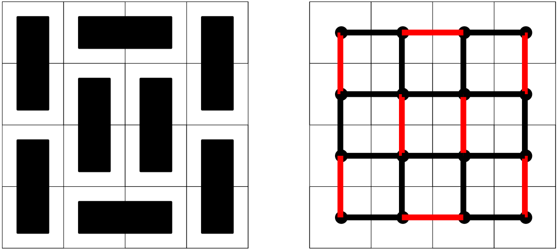
Relevancy to our problem: perfect matchings

Observation: A domino tiling is a perfect matching in the underlying grid graph.



Relevancy to our problem: perfect matchings

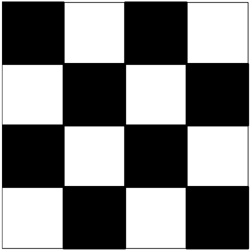
Observation: A domino tiling is a perfect matching in the underlying grid graph.

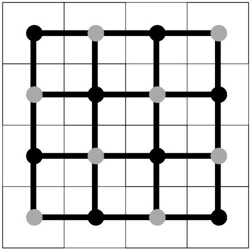


Reason: Edges represent *potential* domino placements (adjacent squares) and all squares must be covered by *exactly* one domino.

Relevancy to our problem: bipartite graphs

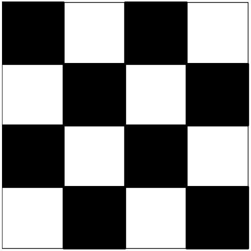
Observation: The underlying grid graph is bipartite.

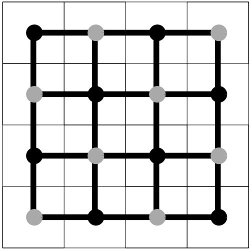




Relevancy to our problem: bipartite graphs

Observation: The underlying grid graph is bipartite.

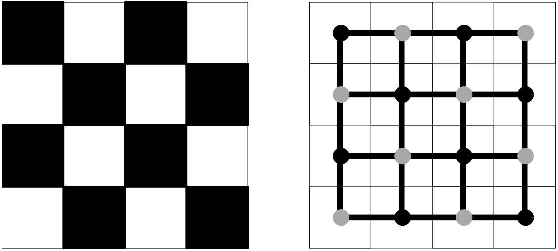




Reason: Color the squares like a chessboard. Take the two vertex parts to be the **black** squares and **white** squares.

Relevancy to our problem: bipartite graphs

Observation: The underlying grid graph is bipartite.



Reason: Color the squares like a chessboard. Take the two vertex parts to be the **black** squares and **white** squares. Edges only connect squares of *opposite* colors, since squares of the *same* color lie along *diagonals*.

Notation

We will use B and W to represent the two vertex parts.

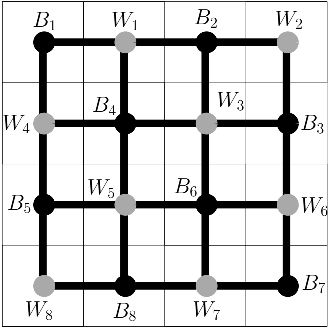
Given m, n the grid graph has mn vertices, so each part has $N := \frac{mn}{2}$ vertices.

Notation

We will use B and W to represent the two vertex parts.

Given m, n the grid graph has mn vertices, so each part has $N := \frac{mn}{2}$ vertices.

We will number the vertices in each part, from 1 to N. Order is irrelevant, but the convention is to snake from the top-left:



Why bother with this formulation?

We can conveniently represent the grid graph as a matrix and exploit its properties.

Definition

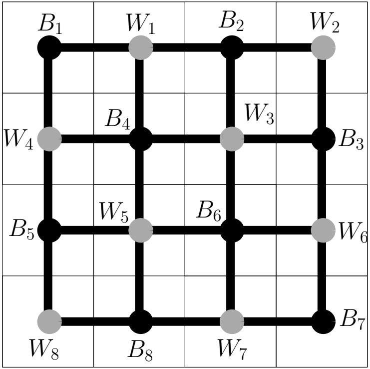
Consider a grid graph G, with $N := \frac{mn}{2}$ vertices in each part. The **adjacency matrix** *A is the $N \times N$ matrix given by*

$$a_{ij} = \begin{cases} 1 & \textit{if } \{b_i, w_j\} \textit{ is an edge in } G \\ 0 & \textit{otherwise} \end{cases}$$

This encodes all of the possible domino placements, so exploring its properties should yield some insight to our problem.

An example adjacency matrix

Recall the 4×4 board and grid graph and construct its corresponding adjacency matrix:



$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

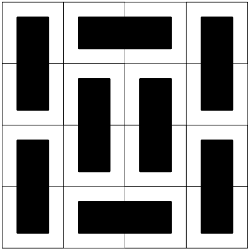
What does a perfect matching look like in A?

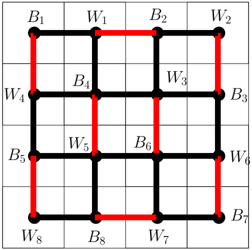
Since B and W each have N labeled vertices, a perfect matching is completely characterized by a **permutation** of $\{1, 2, \ldots, N\}$.

What does a perfect matching look like in A?

Since B and W each have N labeled vertices, a perfect matching is completely characterized by a **permutation** of $\{1, 2, \ldots, N\}$.

Example: Recall this tiling/matching in the 4×4 board:





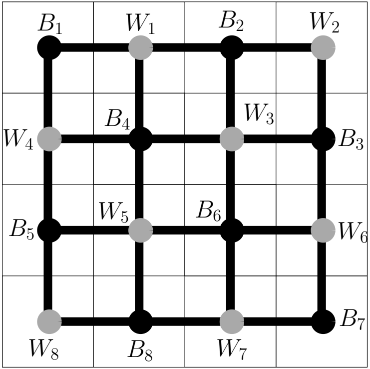
This corresponds to the permutation $(4, 1, 2, 5, 8, 3, 6, 7)$ on $\{1, 2, \ldots, 8\}$. It encodes which w_j is adjacent to each b_i.

This does **not** work the other way!

An arbitrary permutation on $\{1, 2, \ldots, N\}$ does **not** necessarily represent a perfect matching in G, though.

This does **not** work the other way!

An arbitrary permutation on $\{1, 2, \ldots, N\}$ does **not** necessarily represent a perfect matching in G, though.



Example:
$(1, 2, 6, 4, 3, 7, 8, 5)$

Notice that $\{b_5, w_3\}$ and $\{b_7, w_8\}$ are not edges in G (those squares are far apart on the board) so this is not a perfect matching and, thus, not a tiling.

Counting tilings via permutations

Recall that S_N is the set of all permuations of $\{1, 2, \ldots, N\}$. (In fact, it is the *symmetric group* on N elements.)

Given $\pi \in S_N$, does π correspond to a perfect matching in G? *Only* if all of the necessary edges represented by π are, indeed, present in G.

Counting tilings via permutations

Recall that S_N is the set of all permuations of $\{1, 2, \ldots, N\}$. (In fact, it is the *symmetric group* on N elements.)

Given $\pi \in S_N$, does π correspond to a perfect matching in G? *Only* if all of the necessary edges represented by π are, indeed, present in G.

This requires all of the entries $a_{i,\pi(i)}$ to be 1, not 0.

Counting tilings via permutations

Recall that S_N is the set of all permuations of $\{1, 2, \ldots, N\}$. (In fact, it is the *symmetric group* on N elements.)

Given $\pi \in S_N$, does π correspond to a perfect matching in G? *Only* if all of the necessary edges represented by π are, indeed, present in G.

This requires all of the entries $a_{i,\pi(i)}$ to be 1, not 0.

This requires $a_{1,\pi(1)} \cdot a_{2,\pi(2)} \cdots a_{N,\pi(N)} = 1$.

If any such edge is not present, its entry in A will be 0, so the product will be 0.

Counting tilings via the adjacency matrix

Accordingly,

$$T(m,n) = \sum_{\pi \in S_N} a_{1,\pi(1)} \cdot a_{2,\pi(2)} \cdots a_{N,\pi(N)}$$

A permutation that corresponds to a matching in G contributes a 1 to the sum, a permutation that does not correspond to a matching contributes a 0.

Counting tilings via the adjacency matrix

Accordingly,

$$T(m,n) = \sum_{\pi \in S_N} a_{1,\pi(1)} \cdot a_{2,\pi(2)} \cdots a_{N,\pi(N)}$$

A permutation that corresponds to a matching in G contributes a 1 to the sum, a permutation that does not correspond to a matching contributes a 0.

Does this look familiar ...?

Definition

Given an $N \times N$ matrix A, the **permanent** *of A is*

$$\operatorname{per}(A) = \sum_{\pi \in S_N} a_{1,\pi(1)} \cdot a_{2,\pi(2)} \cdots a_{N,\pi(N)}$$

Definition

Given an $N \times N$ *matrix* A*, the* **permanent** *of* A *is*

$$\text{per}(A) = \sum_{\pi \in S_N} a_{1,\pi(1)} \cdot a_{2,\pi(2)} \cdots a_{N,\pi(N)}$$

and the **determinant** *of* A *is*

$$\det(A) = \sum_{\pi \in S_N} \text{sgn}(\pi) \cdot a_{1,\pi(1)} \cdot a_{2,\pi(2)} \cdots a_{N,\pi(N)}$$

where $\text{sgn}(\pi)$ *is* ± 1*, depending on its parity (the number of transpositions required to return* π *to the Identity).*

So . . . are we done?

Given m, n, simply find A and compute $\text{per}(A)$.

So . . . are we done?

Given m, n, simply find A and compute $\text{per}(A)$.

The problem: Computing permanents is *hard!*

So . . . are we done?

Given m, n, simply find A and compute $\mathrm{per}(A)$.

The problem: Computing permanents is *hard!*

Even when the entries are just 0/1 (like we have), computing the permanent is **#P-complete**.

Computational complexity

NP problems are *decision* problems whose proposed answers can be evaluated in polynomial time. For example:

- Given a set of integers, is there a subset whose sum is 0?
- Given a conjuctive normal form formula, is there an assignment of Boolean values that makes the statement evaluate to True?

Computational complexity

NP problems are *decision* problems whose proposed answers can be evaluated in polynomial time. For example:

- Given a set of integers, is there a subset whose sum is 0?
- Given a conjuctive normal form formula, is there an assignment of Boolean values that makes the statement evaluate to True?

#P problems are the *counting* versions of those decision problems in **NP**. Of course, these problems are *harder* to solve!

- Given a set of integers, how many subsets have sum 0?
- Given a conjuctive normal form formula, how many Boolean assignments make the statement True?

Computational complexity

A problem is **#P-complete** if it is in **#P** and every other problem in **#P** can be reduced to it by a polynomial-time counting reduction.

Computational complexity

A problem is **#P-complete** if it is in **#P** and every other problem in **#P** can be reduced to it by a polynomial-time counting reduction.

Given a bipartite graph with V vertices and E edges, *finding* a perfect matching can be done in $O(VE)$ time. Thus, "Is there a perfect matching?" is a **P** problem. It is *easy*.

Computational complexity

A problem is **#P-complete** if it is in **#P** and every other problem in **#P** can be reduced to it by a polynomial-time counting reduction.

Given a bipartite graph with V vertices and E edges, *finding* a perfect matching can be done in $O(VE)$ time. Thus, "Is there a perfect matching?" is a **P** problem. It is *easy*.

However, "How *many* perfect matchings are there?" is **#P-complete**. It is *hard*.

This was proven in 1979 by Valiant. In his paper, he introduced the terms **#P** and **#P-complete**.

Computational complexity

"Thus, if the permanent can be computed in polynomial time by any method, then $FP = \#P$, which is an even stronger statement than $P = NP$." [4]

Computational complexity

"Thus, if the permanent can be computed in polynomial time by any method, then $FP = \#P$, which is an even stronger statement than $P = NP$." [4]

However, computing a **determinant** is easy! Algorithms exist that can compute $\det(A)$ in $O(N^3)$ time.

This is because the determinant has some nice algebraic properties that the permanent does not share.

Computational complexity

"Thus, if the permanent can be computed in polynomial time by any method, then $FP = \#P$, which is an even stronger statement than $P = NP$." [4]

However, computing a **determinant** is easy! Algorithms exist that can compute $\det(A)$ in $O(N^3)$ time.

This is because the determinant has some nice algebraic properties that the permanent does not share.

New goal: Find a matrix $\hat{A}$ such that $|\det(\hat{A})| = \text{per}(A)$, then compute $\det(\hat{A})$. As long as this is done in polynomial-time, we will have solved the overall problem in polynomial-time.

Definition: weighting the edges

Definition

A **signing** *of G is an assignment of ± 1 weights to the edges:*

$$\sigma : E(G) \to \{-1, +1\}$$

The **signed adjacency matrix** A^σ *is given by*

$$a_{ij}^\sigma = \begin{cases} \sigma\left(\{b_i, w_j\}\right) & \textit{if } \{b_i, w_j\} \textit{ is an edge in } G \\ 0 & \textit{otherwise} \end{cases}$$

Definition: weighting the edges

Definition

A **signing** *of G is an assignment of ± 1 weights to the edges:*

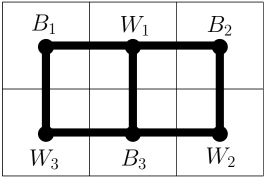
$$\sigma : E(G) \to \{-1, +1\}$$

The **signed adjacency matrix** *A^σ is given by*

$$a_{ij}^{\sigma} = \begin{cases} \sigma\left(\{b_i, w_j\}\right) & \text{if } \{b_i, w_j\} \text{ is an edge in } G \\ 0 & \text{otherwise} \end{cases}$$

If such a σ satisfies the equation $\operatorname{per}(A) = |\det(A^\sigma)|$, then we say σ is a **Kasteleyn signing** *of G.*

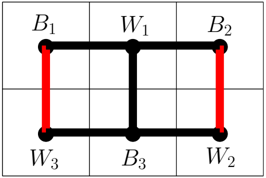
An example: the 2×3 grid graph



$$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Notice $\text{per}(A) = \underbrace{1}_{123} + \underbrace{0}_{132} + \underbrace{0}_{213} + \underbrace{0}_{231} + \underbrace{1}_{312} + \underbrace{1}_{321} = 3.$

An example: the 2×3 grid graph



$$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Notice $\text{per}(A) = \underbrace{1}_{123} + \underbrace{0}_{132} + \underbrace{0}_{213} + \underbrace{0}_{231} + \underbrace{1}_{312} + \underbrace{1}_{321} = 3.$

Weight $\{b_1, w_3\}$ and $\{b_2, w_2\}$ with -1, all others +1. Then,

$A^\sigma = \begin{bmatrix} 1 & 0 & -1 \\ 1 & -1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ and $\det(A^\sigma) = \begin{vmatrix} -1 & 0 \\ 1 & 1 \end{vmatrix} - \begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix} = -3$

A non-example: $K_{3,3}$

Fact: There is *no* Kasteleyn signing of $K_{3,3}$.

A non-example: $K_{3,3}$

Fact: There is *no* Kasteleyn signing of $K_{3,3}$.

Proof.

Notice that $\text{per}(A) = 6$ here, because all entries are 1, and

$$\begin{aligned}\det(A^\sigma) = &\; a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} \\ &- a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} - a_{13}a_{22}a_{31}\end{aligned}$$

A non-example: $K_{3,3}$

Fact: There is *no* Kasteleyn signing of $K_{3,3}$.

Proof.

Notice that $\text{per}(A) = 6$ here, because all entries are 1, and

$$\begin{aligned}\det(A^\sigma) = & \ a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} \\ & - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} - a_{13}a_{22}a_{31}\end{aligned}$$

Let's make these all, say, +1. WOLOG $a_{11} = +1$. Then either a_{22}, a_{33} both +1 or both -1.

If both +1, we get a_{23}, a_{32} and a_{12}, a_{21} and a_{13}, a_{31} have opposite signs. ※

If both -1, we get a_{23}, a_{32} have opposite while a_{12}, a_{21} and a_{13}, a_{31} have same signs. ※ □

Informal statement and proof strategy

Theorem

Every grid graph arising from an $m \times n$ rectangular board has a Kasteleyn signing and we can find one efficiently.

More formal statement forthcoming.

Informal statement and proof strategy

Theorem

Every grid graph arising from an $m \times n$ rectangular board has a Kasteleyn signing and we can find one efficiently.

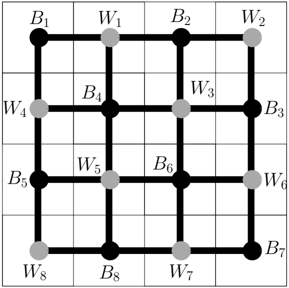
More formal statement forthcoming.

Proof strategy: Lemma 1 provides a *sufficient* condition for a signing to be Kasteleyn. Lemma 2 provides a more specific version of this condition that applies to our grid graphs. The Theorem follows from these two and an algorithm for *building in* the condition of Lemma 2 to a signing.

2-connectivity and planarity

A graph is **planar** if it can be drawn on the plane with no edges crossing.

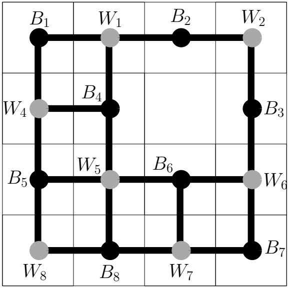
Notice our grid graphs are planar because the rectangular boards are, too. We can just draw the graph on the board!



2-connectivity and planarity

A graph is **2-connected** if it is connected and the removal of any vertex does *not* disconnect the graph.

Notice our grid graphs are 2-connected because even after removing a square, we can connected any two squares with a path of alternating colors; we just might have to "go around" the hole.



Formal statement

Theorem

Let G be a bipartite, planar, 2-connected graph. Then G has a Kasteleyn signing that can be found in polynomial-time (in N).

Corollary

$T(m,n)$ can be computed in polynomial-time (in mn).

Formal statement

Theorem

Let G be a bipartite, planar, 2-connected graph. Then G has a Kasteleyn signing that can be found in polynomial-time (in N).

Corollary

$T(m,n)$ can be computed in polynomial-time (in mn).

Note: The proof will provide an implementable algorithm that is obviously polynomial-time. Matoušek notes that "with some more work" one can find a *linear*-time algorithm.

Formal statement

Theorem

Let G be a bipartite, planar, 2-connected graph. Then G has a Kasteleyn signing that can be found in polynomial-time (in N).

Corollary

$T(m,n)$ can be computed in polynomial-time (in mn).

Note: The proof will provide an implementable algorithm that is obviously polynomial-time. Matoušek notes that "with some more work" one can find a *linear*-time algorithm.

Note: The *bipartite* and *2-connected* assumptions can be removed, with effort, but *planarity* is essential.

Definitions: cycles and signs

Definition

A **cycle** *in G is a sequence of vertices and edges that returns to the same vertex. (It does not need to use all vertices in G.)*

A cycle C is **evenly-placed** *if G has a perfect matching outside of C (i.e. with all edges and vertices of C removed.)*

Notice any cycle in a bipartite graph has *even length*.

Definitions: cycles and signs

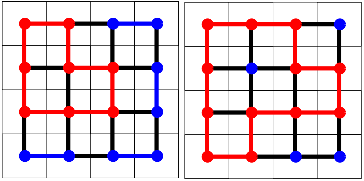
Definition

A **cycle** *in G is a sequence of vertices and edges that returns to the same vertex. (It does not need to use all vertices in G.)*

A cycle C is **evenly-placed** *if G has a perfect matching outside of C (i.e. with all edges and vertices of C removed.)*

Notice any cycle in a bipartite graph has *even length*.

Examples: An evenly-placed and not evenly-placed cycle.



Definitions: cycles and signs

Definition

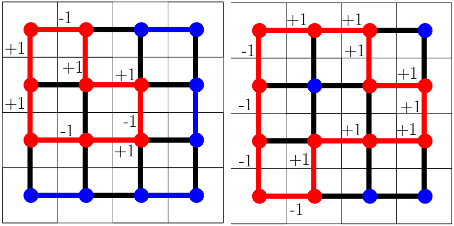
Given σ on G, a cycle C is **properly-signed** *if its length matches the weights of its edges appropriately:*
If $|C| = 2\ell$, then the number of negative edges on C (call it n_C) should have opposite parity of ℓ, i.e. $n_C \equiv \ell - 1 \pmod 2$.

Definitions: cycles and signs

Definition

Given σ on G, a cycle C is **properly-signed** *if its length matches the weights of its edges appropriately:*
If $|C| = 2\ell$, then the number of negative edges on C (call it n_C) should have opposite parity of ℓ, i.e. $n_C \equiv \ell - 1 \pmod 2$.

Examples: A properly-signed and not properly-signed cycle.



Statement

Lemma 1

If every evenly-placed cycle in G is properly-signed, then σ is a Kasteleyn signing.

Proof strategy: We will define the *sign* of a perfect matching. To make sure σ is Kasteleyn, we require all perfect matchings to have the same sign. The symmetric difference of two matchings is a disjoint union of evenly-placed cycles. Since those are properly-signed, we can make a claim about the signs of the permutations corresponding to matchings.

Proof: the sign of a matching

Take σ and suppose every evenly-placed cycle is properly-signed. For any perfect matching M, define

$$\text{sgn}(M) := \text{sgn}(\pi) a^{\sigma}_{1,\pi(1)} a^{\sigma}_{2,\pi(2)} \cdots a^{\sigma}_{N,\pi(N)} = \text{sgn}(\pi) \prod_{e \in M} \sigma(e)$$

Notice this is the corresponding term in the formula for $\det(A)$.

Proof: the sign of a matching

Take σ and suppose every evenly-placed cycle is properly-signed. For any perfect matching M, define

$$\mathrm{sgn}(M) := \mathrm{sgn}(\pi) a^{\sigma}_{1,\pi(1)} a^{\sigma}_{2,\pi(2)} \cdots a^{\sigma}_{N,\pi(N)} = \mathrm{sgn}(\pi) \prod_{e \in M} \sigma(e)$$

Notice this is the corresponding term in the formula for $\det(A)$. To ensure σ is Kasteleyn, we need all matchings to have the *same sign*, so that $\det(A)$ is a sum of all +1s or -1s.

Now, take two arbitrary perfect matchings M, M'.
Goal: Show $\mathrm{sgn}(M) = \mathrm{sgn}(M')$.

Proof: the "product" of two matchings

To achieve this, it suffices to show $\text{sgn}(M)\,\text{sgn}(M') = 1$.

Proof: the "product" of two matchings

To achieve this, it suffices to show $\operatorname{sgn}(M)\operatorname{sgn}(M') = 1$. Notice

$$\begin{aligned}\operatorname{sgn}(M)\operatorname{sgn}(M') &= \operatorname{sgn}(\pi)\operatorname{sgn}(\pi')\left(\prod_{e\in M}\sigma(e)\right)\left(\prod_{e\in M'}\sigma(e)\right)\\ &= \operatorname{sgn}(\pi)\operatorname{sgn}(\pi')\prod_{e\in M\Delta M'}\sigma(e)\\ &= \operatorname{sgn}(\pi)\operatorname{sgn}(\pi')\cdot(-1)^L\end{aligned}$$

because any edge common to both contributes $\sigma(e)^2 = 1$, so we only care about the edges belonging to *exactly* one matching.

Proof: the "product" of two matchings

To achieve this, it suffices to show $\operatorname{sgn}(M)\operatorname{sgn}(M')=1$. Notice

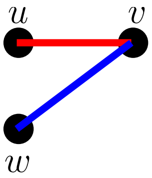
$$\begin{aligned}\operatorname{sgn}(M)\operatorname{sgn}(M') &= \operatorname{sgn}(\pi)\operatorname{sgn}(\pi')\left(\prod_{e\in M}\sigma(e)\right)\left(\prod_{e\in M'}\sigma(e)\right)\\ &= \operatorname{sgn}(\pi)\operatorname{sgn}(\pi')\prod_{e\in M\Delta M'}\sigma(e)\\ &= \operatorname{sgn}(\pi)\operatorname{sgn}(\pi')\cdot(-1)^L\end{aligned}$$

because any edge common to both contributes $\sigma(e)^2=1$, so we only care about the edges belonging to *exactly* one matching.

Goal: Show $\operatorname{sgn}(\pi)=\operatorname{sgn}(\pi')\cdot(-1)^L$, so $\operatorname{sgn}(M)\operatorname{sgn}(M')=1$.

Proof: $M \Delta M'$ is a disjoint union of cycles

Take any vertex u. Find its neighbor v in M.
Find the neighbor w of v in M'.



Proof: $M\Delta M'$ is a disjoint union of cycles

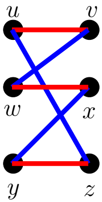
Take any vertex u. Find its neighbor v in M.
Find the neighbor w of v in M'.

If $w = u$ then $\{u, v\}$ is an edge in *both* matchings, so $\{u, v\} \notin M\Delta M'$.

Proof: $M \Delta M'$ is a disjoint union of cycles

Take any vertex u. Find its neighbor v in M.
Find the neighbor w of v in M'.

If $w \neq u$, then repeat this process, alternately finding the next neighbor from M and then M'. Since G is finite, this terminates and closes a cycle.

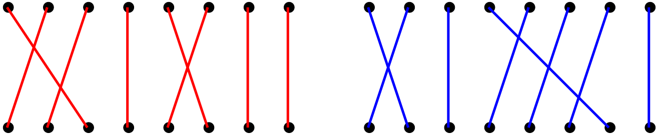


(Note: this cannot close back on itself "internally" since these are *perfect* matchings.)

Repeat on an unused vertex.

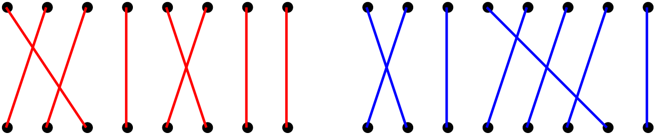
Proof: $M \Delta M'$ is a disjoint union of cycles

Example:
Consider these two matchings on 8 vertices:

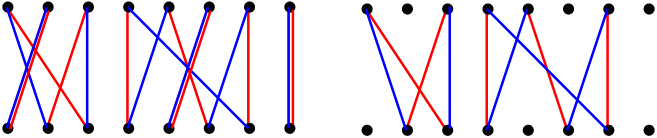


Proof: $M \Delta M'$ is a disjoint union of cycles

Example:
Consider these two matchings on 8 vertices:



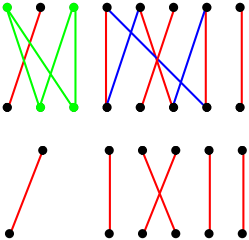
Overlay them and remove common edges to obtain $M \Delta M'$:



Proof: the cycles of $M \Delta M'$ are evenly-placed

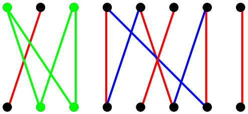
Consider removing such a cycle C from the graph. (Note: its *vertices* are removed, too.)

We can use the edges of, say, M that were not removed. That is, $M - (M \Delta M')$ is a perfect matching on $G - C$.

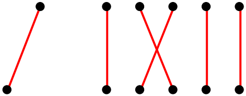


Proof: the cycles of $M\Delta M'$ are evenly-placed

Consider removing such a cycle C from the graph. (Note: its *vertices* are removed, too.)



We can use the edges of, say, M that were not removed. That is, $M - (M\Delta M')$ is a perfect matching on $G - C$.



Thus, all such cycles are evenly-placed, so they are **properly-signed**, by assumption.

This information will help us complete the proof.

Proof: the signs on the cycles

Say $M \Delta M'$ has k cycles, with lengths $|C_i| = 2\ell_i$.
Properly-signed $\implies n_{C_i} \equiv \ell_i - 1 \pmod{2}$ (# of neg. edges)

Proof: the signs on the cycles

Say $M \Delta M'$ has k cycles, with lengths $|C_i| = 2\ell_i$.
Properly-signed $\implies n_{C_i} \equiv \ell_i - 1 \pmod 2$ (# of neg. edges)
Thus, $\prod_{e \in C_i} \sigma(e) = (-1)^{n_{C_i}} = (-1)^{\ell_i - 1}$, and so

Proof: the signs on the cycles

Say $M \Delta M'$ has k cycles, with lengths $|C_i| = 2\ell_i$.
Properly-signed $\implies n_{C_i} \equiv \ell_i - 1 \pmod 2$ (# of neg. edges)
Thus, $\prod_{e \in C_i} \sigma(e) = (-1)^{n_{C_i}} = (-1)^{\ell_i - 1}$, and so

$$\begin{aligned}
\operatorname{sgn}(M)\operatorname{sgn}(M') &= \operatorname{sgn}(\pi)\operatorname{sgn}(\pi') \prod_{i=1}^{k} \left[\prod_{e \in C_i} \sigma(e) \right] \\
&= \operatorname{sgn}(\pi)\operatorname{sgn}(\pi') \prod_{i=1}^{k} (-1)^{\ell_i - 1} \\
&= \operatorname{sgn}(\pi)\operatorname{sgn}(\pi') \cdot (-1)^{L} \\
&\qquad \text{where } L := (\ell_1 - 1) + (\ell_2 - 1) + \cdots + (\ell_k - 1)
\end{aligned}$$

Proof: the signs on the cycles

Say $M \Delta M'$ has k cycles, with lengths $|C_i| = 2\ell_i$.
Properly-signed $\implies n_{C_i} \equiv \ell_i - 1 \pmod 2$ (# of neg. edges)
Thus, $\prod_{e \in C_i} \sigma(e) = (-1)^{n_{C_i}} = (-1)^{\ell_i - 1}$, and so

$$\begin{aligned} \operatorname{sgn}(M)\operatorname{sgn}(M') &= \operatorname{sgn}(\pi)\operatorname{sgn}(\pi') \prod_{i=1}^{k} \left[\prod_{e \in C_i} \sigma(e) \right] \\ &= \operatorname{sgn}(\pi)\operatorname{sgn}(\pi') \prod_{i=1}^{k} (-1)^{\ell_i - 1} \\ &= \operatorname{sgn}(\pi)\operatorname{sgn}(\pi') \cdot (-1)^{L} \\ &\qquad \text{where } L := (\ell_1 - 1) + (\ell_2 - 1) + \cdots + (\ell_k - 1) \end{aligned}$$

Proof: π and π' differ by L transpositions

Claim: We can morph π into π' by considering these cycles and identifying L transpositions.

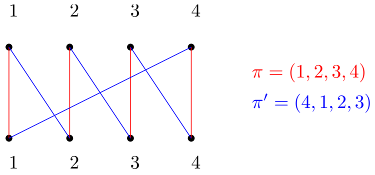
Take C_i. We will identify $\ell_i - 1$ transpositions that will make π and π' identical on the vertices of C_i.

Proof: π and π' differ by L transpositions

Claim: We can morph π into π' by considering these cycles and identifying L transpositions.

Take C_i. We will identify $\ell_i - 1$ transpositions that will make π and π' identical on the vertices of C_i.

Relabel vertices so π and π' are ordered permutations on $\{1, 2, \ldots, \ell_i\}$. Since C_i is a cycle, *no* positions are identical.



Proof: π and π' differ by L transpositions

Algorithm: At step $t = 1, \dots, \ell_i - 1$:

Proof: π and π' differ by L transpositions

Algorithm: At step $t = 1, \dots, \ell_i - 1$:

Proof: π and π' differ by L transpositions

Algorithm: At step $t = 1, \ldots, \ell_i - 1$:

- Identify j, k such that $\pi(j) = \pi'(k) = t$.

Proof: π and π' differ by L transpositions

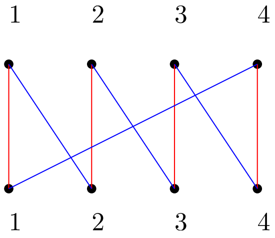
Algorithm: At step $t = 1, \ldots, \ell_i - 1$:

- Identify j, k such that $\pi(j) = \pi'(k) = t$.
- Swap positions j and k in π

Proof: π and π' differ by L transpositions

Algorithm: At step $t = 1, \ldots, \ell_i - 1$:

- Identify j, k such that $\pi(j) = \pi'(k) = t$.
- Swap positions j and k in π



$\pi = (1, 2, 3, 4)$ $\pi(1) = 1$

$\pi' = (4, 1, 2, 3)$ $\pi'(2) = 1$

Swap positions 1 and 2 in π

Proof: π and π' differ by L transpositions

Claim: Such a step is always possible, and it will introduce *exactly* one identical position between π and π'; namely, they now agree in the position where t appears.

Proof: π and π' differ by L transpositions

Claim: Such a step is always possible, and it will introduce *exactly* one identical position between π and π'; namely, they now agree in the position where t appears.

The only issue would occur if we somehow introduced *two* identical positions when we made this swap.

Proof: π and π' differ by L transpositions

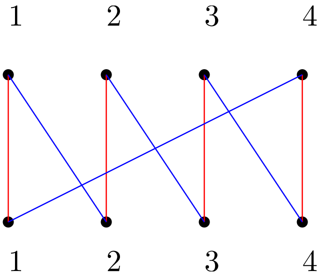
Claim: Such a step is always possible, and it will introduce *exactly* one identical position between π and π'; namely, they now agree in the position where t appears.

The only issue would occur if we somehow introduced *two* identical positions when we made this swap.

This only happens if $\pi(j) = \pi'(k) = t$ and also $\pi(k) = \pi'(j)$. This means $(j, \pi(k), k, \pi(k))$ was a 4-cycle to begin with.

Proof: π and π' differ by L transpositions

For illustration's sake, here is how that process would play out:

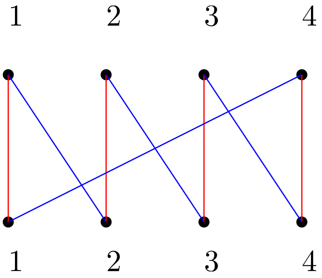


$\pi = (1, 2, 3, 4)$

$\pi' = (4, 1, 2, 3)$

Proof: π and π' differ by L transpositions

For illustration's sake, here is how that process would play out:



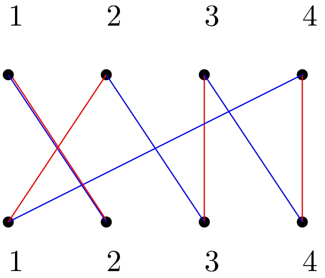
$\pi = (1, 2, 3, 4)$ $\pi(1) = 1$

$\pi' = (4, 1, 2, 3)$ $\pi'(2) = 1$

Swap positions 1 and 2 in π

Proof: π and π' differ by L transpositions

For illustration's sake, here is how that process would play out:

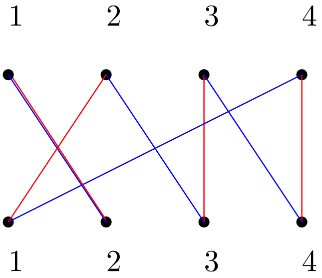


$\pi = (2, 1, 3, 4)$

$\pi' = (4, 1, 2, 3)$

Proof: π and π' differ by L transpositions

For illustration's sake, here is how that process would play out:



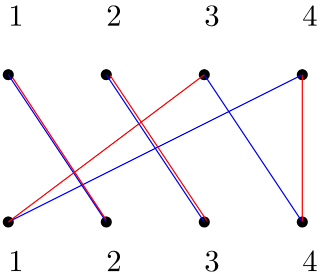
$\pi = (2, 1, 3, 4)$ $\pi(1) = 2$

$\pi' = (4, 1, 2, 3)$ $\pi'(3) = 2$

Swap positions 1 and 3 in π

Proof: π and π' differ by L transpositions

For illustration's sake, here is how that process would play out:

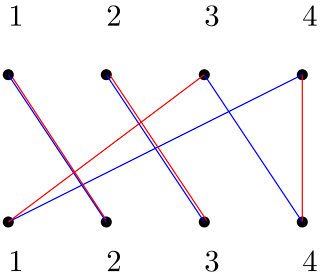


$\pi = (3, 1, 2, 4)$

$\pi' = (4, 1, 2, 3)$

Proof: π and π' differ by L transpositions

For illustration's sake, here is how that process would play out:



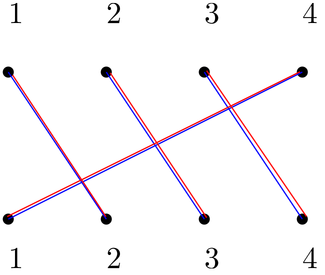
$\pi = (3, 1, 2, 4)$ $\pi(1) = 3$

$\pi' = (4, 1, 2, 3)$ $\pi'(4) = 3$

Swap positions 1 and 4 in π

Proof: π and π' differ by L transpositions

For illustration's sake, here is how that process would play out:



$\pi = (4, 1, 2, 3)$

$\pi' = (4, 1, 2, 3)$

Now $\pi = \pi'$

Proof: wrapping up

Since π and π' differ by L transpositions,
$\text{sgn}(\pi) = \text{sgn}(\pi') \cdot (-1)^L$.

Proof: wrapping up

Since π and π' differ by L transpositions,
$\text{sgn}(\pi) = \text{sgn}(\pi') \cdot (-1)^L$.

Plugging this into the formula, we have $\text{sgn}(M)\,\text{sgn}(M') = 1$.

Proof: wrapping up

Since π and π' differ by L transpositions, $\text{sgn}(\pi) = \text{sgn}(\pi') \cdot (-1)^L$.

Plugging this into the formula, we have $\text{sgn}(M)\,\text{sgn}(M') = 1$.

Thus, all perfect matchings in G have the same sign, so they contribute the same term (-1 or +1) to the determinant formula.

Therefore, $|\det(A^\sigma)| = \text{per}(A)$, and σ is Kasteleyn. □

Proof: wrapping up

Since π and π' differ by L transpositions,
$\operatorname{sgn}(\pi) = \operatorname{sgn}(\pi') \cdot (-1)^L$.

Plugging this into the formula, we have $\operatorname{sgn}(M)\operatorname{sgn}(M') = 1$.

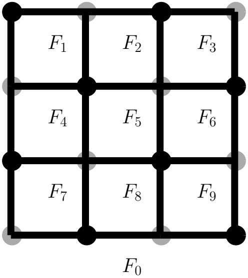
Thus, all perfect matchings in G have the same sign, so they contribute the same term (-1 or +1) to the determinant formula.

Therefore, $|\det(A^\sigma)| = \operatorname{per}(A)$, and σ is Kasteleyn. □

We now have a way of more easily checking if a signing is Kasteleyn. The next Lemma helps us check even *more* easily because it exploits the planarity and 2-connectivity of G.

Planar graphs

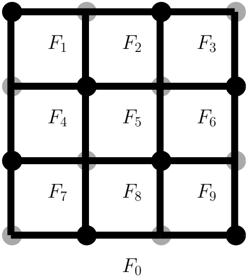
A planar drawing of a graph has **vertices**, **edges**, and **faces**.



There is one *outer face*; the rest are *inner faces*.

Planar graphs

A planar drawing of a graph has **vertices**, **edges**, and **faces**.



There is one *outer face*; the rest are *inner faces*.

Euler's Formula: $V + F = E + 2$

Statement and proof strategy

Lemma

Fix a planar drawing of a bipartite, planar, 2-connected graph G, with signing σ. If the boundary cycle of every inner face is properly-signed, then σ is Kasteleyn.

Statement and proof strategy

Lemma

Fix a planar drawing of a bipartite, planar, 2-connected graph G, with signing σ. If the boundary cycle of every inner face is properly-signed, then σ is Kasteleyn.

Proof strategy: Overall, we invoke Lemma 1. An arbitrary, well-placed cycle C encloses some inner faces. Euler's Formula relates $|C|$ and the lengths of the boundary cycles inside C. The proper-signing of those boundary cycles will tell us C is also properly-signed, so Lemma 1 applies.

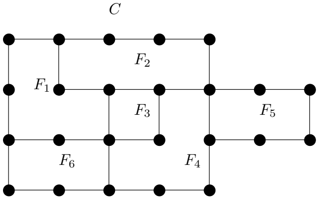
Proof: An evenly-placed cycle encloses inner faces

Let C be an evenly-placed cycle in G. Restrict our attention to the vertices and edges inside and on C.

Proof: An evenly-placed cycle encloses inner faces

Let C be an evenly-placed cycle in G. Restrict our attention to the vertices and edges inside and on C.

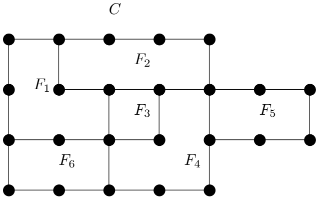
Say we have inner faces $F_1, \ldots, F_k$ with boundary cycles C_i of length $2\ell_i$.



Proof: An evenly-placed cycle encloses inner faces

Let C be an evenly-placed cycle in G. Restrict our attention to the vertices and edges inside and on C.

Say we have inner faces $F_1, \ldots, F_k$ with boundary cycles C_i of length $2\ell_i$.



Counting:

- $V = r + 2\ell$ (where r is the number of vertices *inside* C)
- $E = \frac{1}{2}\left(|C| + |C_1| + |C_2| + \cdots + |C_k|\right) = \ell + \ell_1 + \cdots + \ell_k$
- $F = k + 1$ (including the outer face)

Proof: applying Euler's Formula and assumptions

Euler's Formula $\implies$

$$r + 2\ell + k + 1 = \ell + \ell_1 + \cdots + \ell_k + 2$$

Proof: applying Euler's Formula and assumptions

Euler's Formula $\Longrightarrow$

$$r + 2\ell + k + 1 = \ell + \ell_1 + \cdots + \ell_k + 2$$

C evenly-placed $\Longrightarrow$ r even $\Longrightarrow$

$$2s + \ell - 1 = \ell_1 + \cdots + \ell_k - k$$

Proof: applying Euler's Formula and assumptions

Euler's Formula $\implies$

$$r + 2\ell + k + 1 = \ell + \ell_1 + \cdots + \ell_k + 2$$

C evenly-placed $\implies$ r even $\implies$

$$2s + \ell - 1 = \ell_1 + \cdots + \ell_k - k$$

Reducing mod 2 $\implies$

$$\ell - 1 \equiv \ell_1 + \cdots + \ell_k - k \pmod 2$$

Proof: applying Euler's Formula and assumptions

Euler's Formula $\implies$

$$r + 2\ell + k + 1 = \ell + \ell_1 + \cdots + \ell_k + 2$$

C evenly-placed $\implies$ r even $\implies$

$$2s + \ell - 1 = \ell_1 + \cdots + \ell_k - k$$

Reducing mod 2 $\implies$

$$\ell - 1 \equiv \ell_1 + \cdots + \ell_k - k \pmod 2$$

Goal: Use this to show $n_C \equiv \ell - 1 \pmod 2$.

Proof: negative edges on and inside C

Every edge appears on *exactly two* of the cycles: $C, C_1, \ldots, C_k$

Proof: negative edges on and inside C

Every edge appears on *exactly two* of the cycles: $C, C_1, \ldots, C_k$

$\implies n_C + n_{C_1} + \cdots + n_{C_k}$ is *even*

Proof: negative edges on and inside C

Every edge appears on *exactly two* of the cycles: $C, C_1, \ldots, C_k$

$\implies n_C + n_{C_1} + \cdots + n_{C_k}$ is *even*

$\implies n_C \equiv n_{C_1} + \cdots + n_{C_k} \pmod{2}$

Proof: negative edges on and inside C

Every edge appears on *exactly two* of the cycles: $C, C_1, \ldots, C_k$

$\implies n_C + n_{C_1} + \cdots + n_{C_k}$ is *even*

$\implies n_C \equiv n_{C_1} + \cdots + n_{C_k} \pmod 2$

The C_i are properly-signed $\implies n_{C_i} \equiv \ell_i - 1 \pmod 2$

Proof: negative edges on and inside C

Every edge appears on *exactly two* of the cycles: $C, C_1, \ldots, C_k$

$\implies n_C + n_{C_1} + \cdots + n_{C_k}$ is *even*

$\implies n_C \equiv n_{C_1} + \cdots + n_{C_k} \pmod 2$

The C_i are properly-signed $\implies n_{C_i} \equiv \ell_i - 1 \pmod 2$

Overall, then

$$n_C \equiv (\ell_1 - 1) + \cdots + (\ell_k - 1) \equiv \ell_1 + \cdots + \ell_k - k \equiv \ell - 1 \pmod 2$$

so C is properly-signed, as well! Apply Lemma 1. □

Constructing a signing that satisfies Lemma 2

Take our grid graph G and fix a planar drawing. We will describe a method that constructs a signing σ that guarantees every inner face's boundary cycle is properly-signed. It will do this, essentially, one-by-one for each face (whence polynomial-time).

Constructing a signing that satisfies Lemma 2

Take our grid graph G and fix a planar drawing. We will describe a method that constructs a signing σ that guarantees every inner face's boundary cycle is properly-signed. It will do this, essentially, one-by-one for each face (whence polynomial-time).

Set $G_1 := G$. Obtain G_{i+1} from G_i by deleting an edge e_i that separates an inner face F_i from the outer face.

Constructing a signing that satisfies Lemma 2

Take our grid graph G and fix a planar drawing. We will describe a method that constructs a signing σ that guarantees every inner face's boundary cycle is properly-signed. It will do this, essentially, one-by-one for each face (whence polynomial-time).

Set $G_1 := G$. Obtain G_{i+1} from G_i by deleting an edge e_i that separates an inner face F_i from the outer face.

Eventually, we have G_k with no inner faces.

Constructing a signing that satisfies Lemma 2

Sign the edges remaining arbitrarily (all $+1$, say).

Constructing a signing that satisfies Lemma 2

Sign the edges remaining arbitrarily (all +1, say).

Work backwards, adding $e_{k-1}, e_{k-2}, \ldots, e_1$ back in and choosing their signs.

Constructing a signing that satisfies Lemma 2

Sign the edges remaining arbitrarily (all +1, say).

Work backwards, adding $e_{k-1}, e_{k-2}, \dots, e_1$ back in and choosing their signs.

When e_i is added back in, it is the boundary of only the inner face F_i in G_i. All the other boundary edges of F_i are present, so we have a definitive choice whether $\sigma(e_i) = \pm 1$ to ensure that boundary cycle is properly-signed. □

Constructing a signing that satisfies Lemma 2

Sign the edges remaining arbitrarily (all +1, say).

Work backwards, adding $e_{k-1}, e_{k-2}, \ldots, e_1$ back in and choosing their signs.

When e_i is added back in, it is the boundary of only the inner face F_i in G_i. All the other boundary edges of F_i are present, so we have a definitive choice whether $\sigma(e_i) = \pm 1$ to ensure that boundary cycle is properly-signed. □

(This can't screw up, because once a boundary cycle is *fixed* to be properly-signed, it won't affect the signing of any other cycle. This fixing happens when its *last* boundary edge is added.)

What have we accomplished?

Given m, n we find $T(m, n)$ by the following steps:

What have we accomplished?

Given m, n we find $T(m,n)$ by the following steps:

- Construct the grid graph G for the $m \times n$ board. (Easy)

What have we accomplished?

Given m, n we find $T(m, n)$ by the following steps:

- Construct the grid graph G for the $m \times n$ board. (Easy)
- Take a planar drawing of G. (Easy)

What have we accomplished?

Given m, n we find $T(m, n)$ by the following steps:

- Construct the grid graph G for the $m \times n$ board. (Easy)
- Take a planar drawing of G. (Easy)
- Iteratively remove edges from G until there is only one face. (Not fast, but easy)

What have we accomplished?

Given m, n we find $T(m, n)$ by the following steps:

- Construct the grid graph G for the $m \times n$ board. (Easy)
- Take a planar drawing of G. (Easy)
- Iteratively remove edges from G until there is only one face. (Not fast, but easy)
- Assign weights to remaining edges. Then, add those removed edges back and identify which signs they need. (Not fast, but easy)

What have we accomplished?

Given m, n we find $T(m, n)$ by the following steps:

- Construct the grid graph G for the $m \times n$ board. (Easy)
- Take a planar drawing of G. (Easy)
- Iteratively remove edges from G until there is only one face. (Not fast, but easy)
- Assign weights to remaining edges. Then, add those removed edges back and identify which signs they need. (Not fast, but easy)
- Identify the signed adjacency matrix A^{σ}. (Easy)

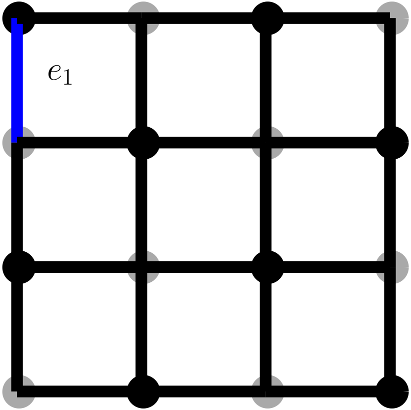
What have we accomplished?

Given m, n we find $T(m, n)$ by the following steps:

- Construct the grid graph G for the $m \times n$ board. (Easy)
- Take a planar drawing of G. (Easy)
- Iteratively remove edges from G until there is only one face. (Not fast, but easy)
- Assign weights to remaining edges. Then, add those removed edges back and identify which signs they need. (Not fast, but easy)
- Identify the signed adjacency matrix A^σ. (Easy)
- Compute $\det(A^\sigma)$. (Computationally fast)

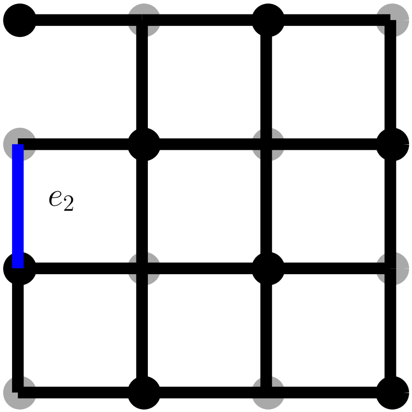
$T(4,4) = ?$

Set $G_1 := G$. Identify e_1 and remove it.



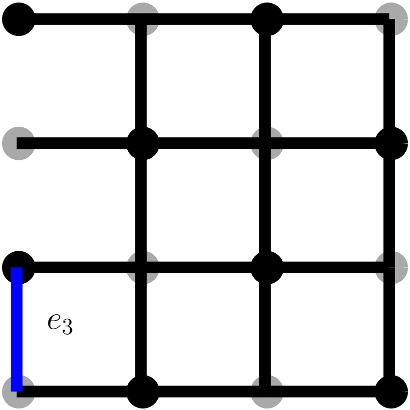
$T(4,4) = ?$

Identify e_2 and remove it.



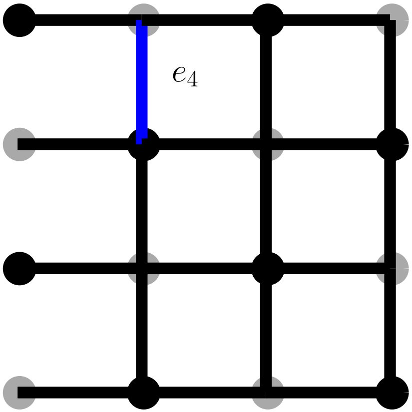
$T(4,4) = ?$

Identify e_3 and remove it.



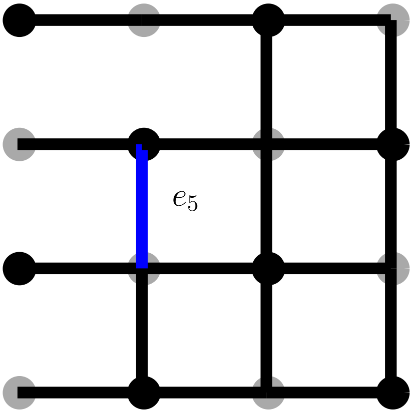
$T(4,4) = ?$

Identify e_4 and remove it.



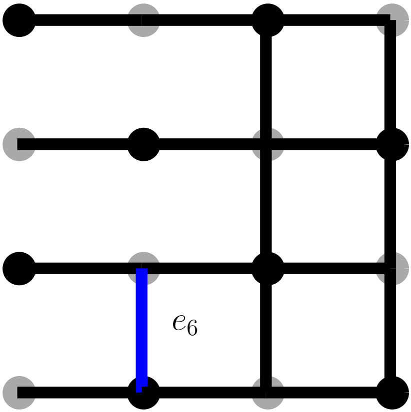
$T(4,4) = ?$

Identify e_5 and remove it.



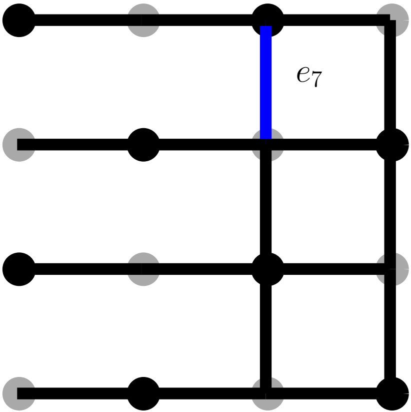
$T(4,4) = ?$

Identify e_6 and remove it.



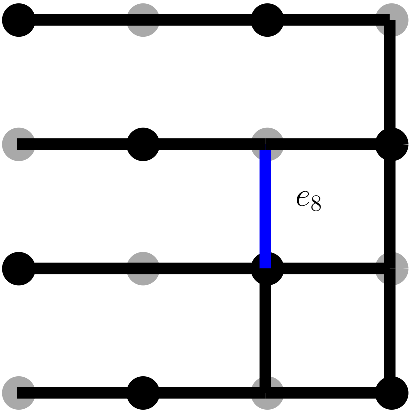
$T(4,4) = ?$

Identify e_7 and remove it.



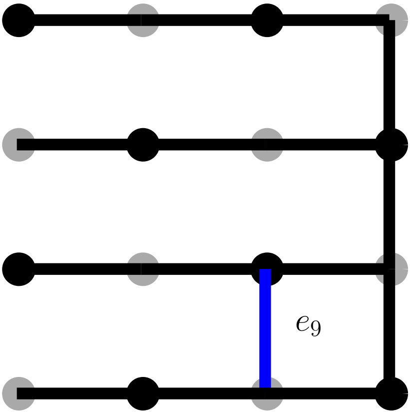
$T(4,4) = ?$

Identify e_8 and remove it.



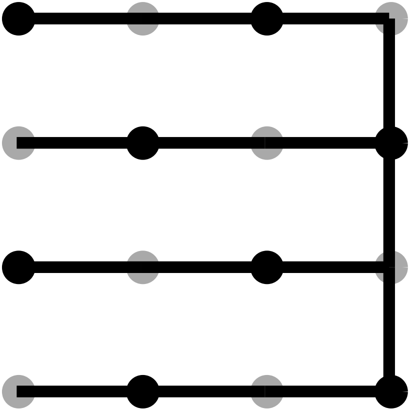
$T(4, 4) = ?$

Identify e_9 and remove it.



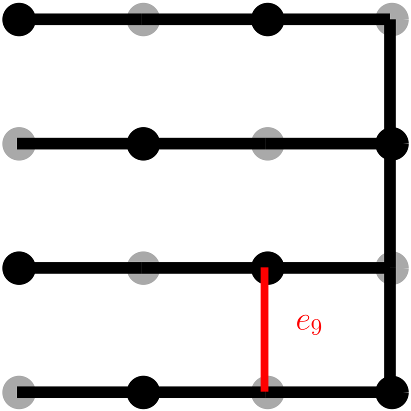
$T(4,4) = ?$

Assign +1 to all remaining edges. (Note: +1 and -1.)



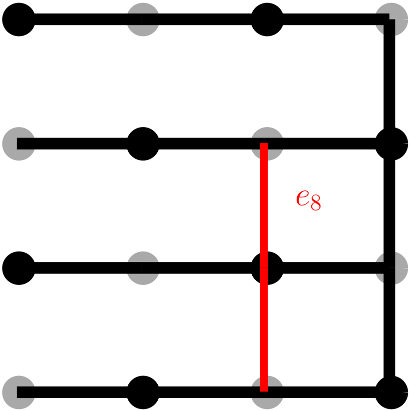
$T(4,4) = ?$

Add e_9 back in. It must be -1.



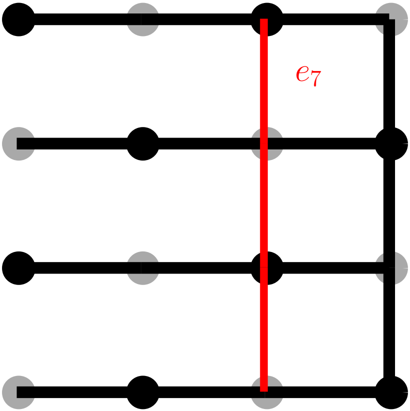
$T(4,4) = ?$

Add e_8 back in. It must be -1.



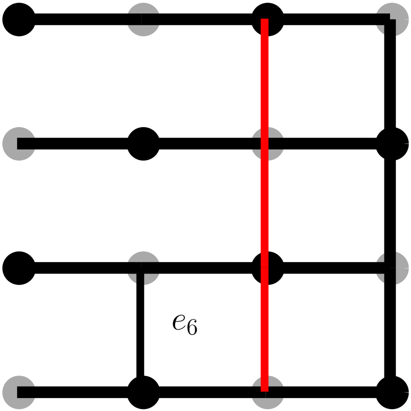
$T(4,4) = ?$

Add e_7 back in. It must be -1.



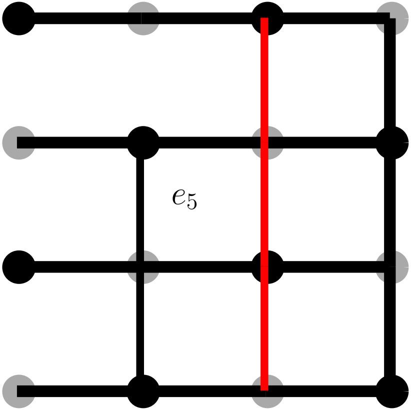
$T(4,4) = ?$

Add e_6 back in. It must be $+1$.



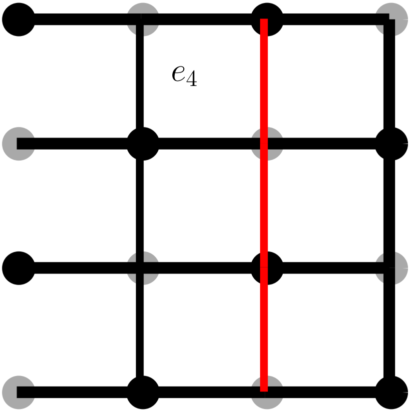
$T(4,4) = ?$

Add e_5 back in. It must be $+1$.



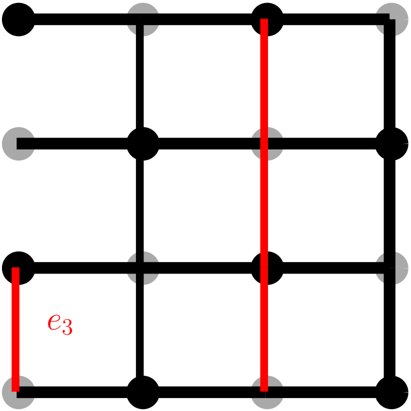
$T(4,4) = ?$

Add e_4 back in. It must be $+1$.



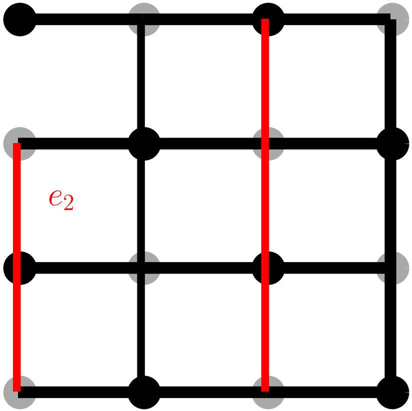
$T(4,4) = ?$

Add e_3 back in. It must be -1.



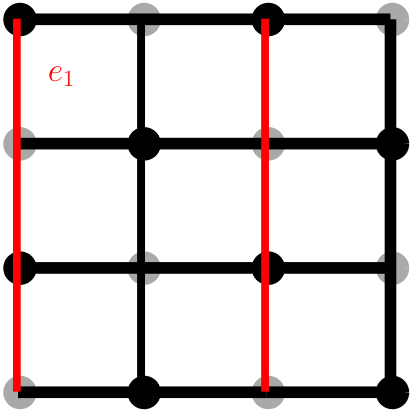
$T(4,4) = ?$

Add e_2 back in. It must be -1.



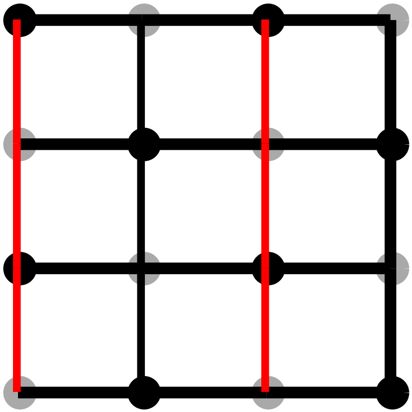
$T(4,4) = ?$

Add e_1 back in. It must be -1.



$T(4,4) = ?$

This is a Kasteleyn signing of G.



$T(4,4) = 36$

$$A^\sigma = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \qquad \det(A^\sigma) = 36 = T(4,4)$$

Crazy product out of nowhere

Amazingly, there is a closed-form solution:

$$T(m,n) = \prod_{k=1}^{m} \prod_{\ell=1}^{n} \left| 2\cos\frac{k\pi}{m+1} + 2\mathrm{i}\cos\frac{\ell\pi}{n+1} \right|$$
$$= \prod_{k=1}^{m} \prod_{\ell=1}^{n} \left(4\cos^2\frac{k\pi}{m+1} + 4\cos^2\frac{\ell\pi}{n+1} \right)^{1/2}$$

Having this shortens the computation time required, of course. Deriving it involves several extra steps.

Cartesian products and eigenvalues

One can show that the adjacency matrices of grid graphs are actually adjacency matrices of the **Cartesian product** of two graphs: a $1 \times n$ row graph and an $m \times 1$ column graph.

The eigenvalues of those matrices are “easily” comptuable.

The determinant of a matrix is the product of its eigenvalues.

Cartesian products and eigenvalues

One can show that the adjacency matrices of grid graphs are actually adjacency matrices of the **Cartesian product** of two graphs: a $1 \times n$ row graph and an $m \times 1$ column graph.

The eigenvalues of those matrices are "easily" comptuable.

The determinant of a matrix is the product of its eigenvalues.

This is explored through a series of problems, whose solutions are also available online [5].

This "ruins the fun" of finding $T(m, n)$ by hand, and doesn't belie any inherent structure/pattern to the problem.

Areas that are being/should be explored

- Hexagonal tilings: closed-form, patterns, etc.
- Random tilings: any regularity?
- Counting perfect matchings in *any* planar graph (Kasteleyn, the Pfaffian method)
- Applications to theoretical physics
- Accounting for isomorphic tilings
- Computational complexity of determinants and permanents
- Enumeration of tilings
- Analyzing closed form: patterns, asymptotics, etc.

References

J. Matoušek.
Thirty-three Miniatures: Mathematical and Algorithmic Applications of Linear Algebra.
AMS, 2010.

P. W. Kasteleyn
The statistics of dimers on a lattic I. The number of dimer arrangements on a quadratic lattice
Physica, **27** (1961) 1209-1225

H. N. V. Temperley, M. E. Fisher
Dimer problem in statistical mechanics — An exact result
Philosophical Magazine, **6** (1961) 1061-1063

Wikipedia
Permanent is sharp-P-complete
http://en.wikipedia.org/wiki/Permanent_is_sharp-P-complete

Aaron Schild
Domino Tilings of a Rectangular Chessboard
http://www.sdmathcircle.org/uploads/Documents/2009-10%20Gauss%2010-10%20Rectangular%20Tiling%20Notes.pdf

THANK YOU

☺