An Experimental Mathematics Approach to Several Combinatorial Problems

Yukun Yao

Department of Mathematics
Rutgers University-New Brunswick

March 17, 2020

Content

- Experimental Mathematics
- Parking Functions
- The Gordian Knot of the C-finite Ansatz
- Analysis of Quicksort Algorithms
- Peaceable Queens Problem
- Summary

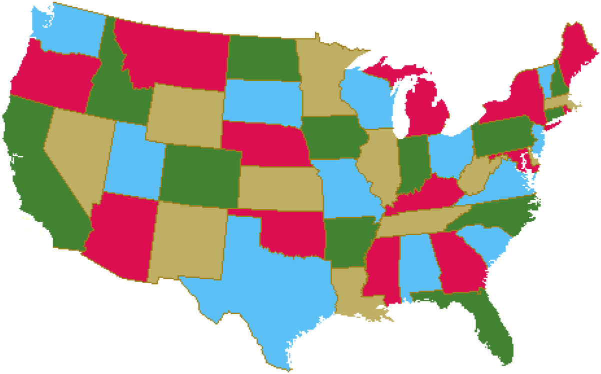
Experimental Mathematics

What is Experimental Mathematics?

Experimental mathematics is an experimental approach to mathematics in which programming and symbolic computation are used to investigate mathematical objects, identify properties and patterns, discover facts and formulas and even automatically prove theorems.

Four Color Theorem

For example, the proof of four color theorem was assisted by computers to check the 1,482 reducible configurations. Without computers, the proof might be impossible.

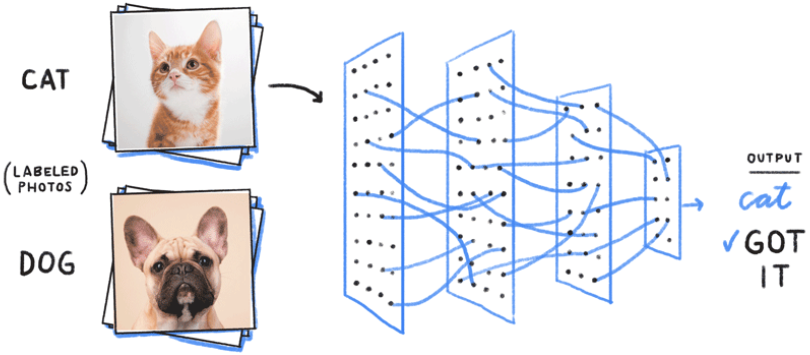


Advantages of Experimental Mathematics approaches

- Efficient
- Easier
- Automatic
- Powerful
- Less error-prone
- Tireless
- And beyond

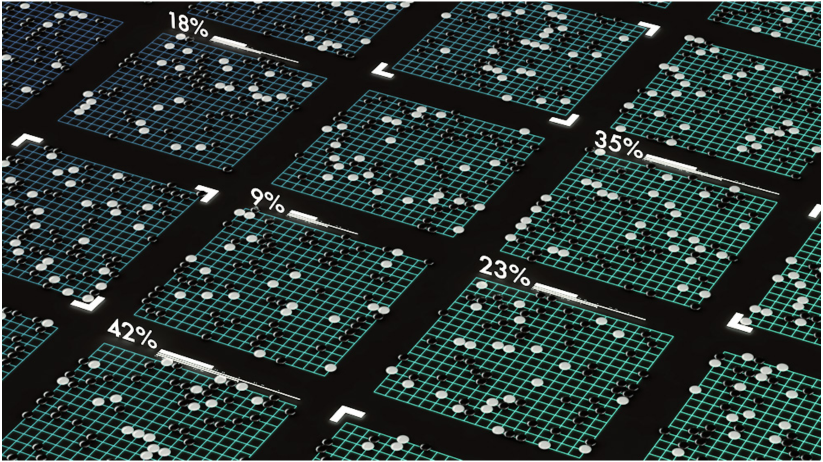
Machine Learning

Machine learning revolutionizes information technology. It can do what humans can.



Machine Learning

It can do better than humans. AlphaGo beat human world champions.



Machine Learning

And it can do what humans can't do or what takes too long to do.

- Detect financial fraud
- Recommend system
- Online search
- Find pattern from big data

Experimental Mathematics

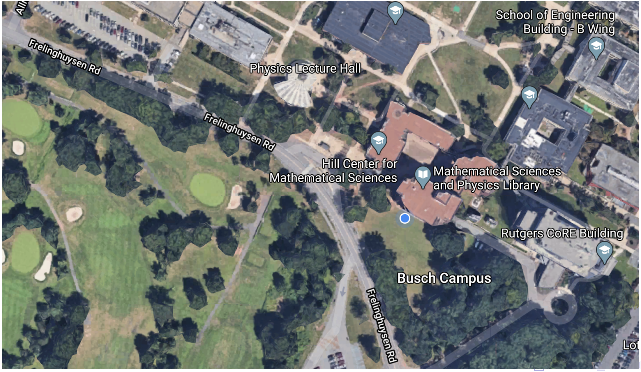
As machine learning revolutionizes information technology, experimental mathematics revolutionizes mathematics. It can

- Look for a pattern
- Test a conjecture
- Utilize data to make a discovery
- Prove theorems automatically
- Provide better tools to maintain and continue building the mathematical skyscraper

Parking Functions

What are parking functions?

In a parallel universe, Frelinghuysen Road is a one-way street (from East to West). There are several parking spaces, say n, on the southern side of Hill Center. At the beginning of today, all the spaces are available. Then n cars come to park one by one, each car i having its favorite parking space number a_i. If all cars can park, we call the preference vector a parking function.



Are they parking functions?

- 978653124
- 123
- 321
- 221
- 222

Definition of parking functions

Definition

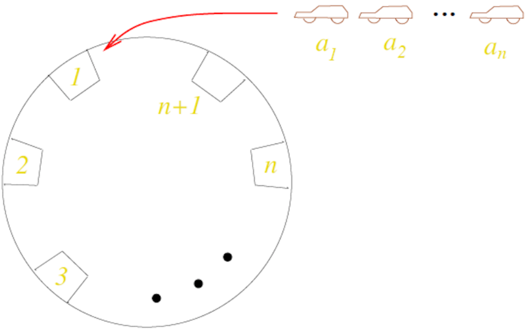
A vector of positive integers $(p_1, \ldots, p_n)$ with $1 \leq p_i \leq n$ is a parking function if its (non-decreasing) sorted version $(p_{(1)}, \ldots, p_{(n)})$ satisfies

$$p_{(i)} \leq i, \quad (1 \leq i \leq n).$$

Number of parking functions

Theorem (Pyke, 1959; Konheim and Weiss, 1966)

Let $f(n)$ be the number of parking functions of length n, then $f(n) = (n+1)^{n-1}$.



Generalization of parking functions: $\hat{k}$-parking function

Definition

A vector of positive integers $(p_1, \ldots, p_n)$ with $1 \le p_i \le n$ is a $\hat{k}$-parking function if its (non-decreasing) sorted version $(p_{(1)}, \ldots, p_{(n)})$ satisfies

$$1 \le p_{(i)} \le ki, \quad (1 \le i \le n).$$

The number of $\hat{k}$-parking functions of length n is $k^n(n+1)^{n-1}$ because any $\hat{k}$-parking function can be written as

$$k(q_1, \ldots, q_n) - (r_1, \ldots, r_n)$$

where (q_i) is a $\hat{1}$-parking function and $0 \le r_i \le k-1$ for each i.

Generalization of parking functions: $\vec{x}$-parking function

Definition

A vector of positive integers $(p_1, \ldots, p_n)$ with $1 \le p_i \le n$ is a $\vec{x}$-parking function, where $\vec{x} \in \mathbb{N}^n$, if its (non-decreasing) sorted version $(p_{(1)}, \ldots, p_{(n)})$ satisfies

$$1 \le p_{(i)} \le \sum_{j=1}^{i} \vec{x}[j], \quad (1 \le i \le n).$$

The number of $\vec{x}$-parking functions of length n obviously depends on $\vec{x}$. When $\vec{x} = (a, b, b, \ldots, b) \in \mathbb{N}^n$, the number is $a(a+nb)^{n-1}$.

Generalization of parking functions: a-parking function

Definition

A vector of positive integers $(p_1, \ldots, p_n)$ with $1 \le p_i \le n$ is an a-parking function if its (non-decreasing) sorted version $(p_{(1)}, \ldots, p_{(n)})$ satisfies

$$1 \le p_{(i)} \le a + i - 1, \quad (1 \le i \le n).$$

We will focus on a-parking functions. And from a-parking functions we can have our experimental mathematics motivated proof of the number of parking functions.

Recurrence relation for a-parking functions

Let $C(n,a)$ be the number of sorted a-parking functions of length n. Consider the number of 1's. If there are k 1's, delete them and consider $a_{k+1}-1,...,a_n-1$. It is an $(a+k-1)$-parking function of length $n-k$. Hence

$$C(n,a)=\sum_{k=0}^{n} C(n-k,a+k-1).$$

Let $P(n,a)$ be the number of a-parking function. With similar argument we have

$$P(n,a)=\sum_{k=0}^{n} \binom{n}{k} P(n-k,a+k-1).$$

$P(n, a) = a(a+n)^{n-1}$

With experimental mathematics and Maple programming,

```
p:=proc(n,a) local k,b:
if n=0 then
RETURN(1)
else
factor(subs(b=a,sum(expand(add(binomial(n,k)*subs(a=a+k-1,p(n-k,a)),k=1..n)),a=1..b))
fi:
end:
```

immediately we will get the list

$$[a, a(a+2), a(a+3)^2, a(a+4)^3, a(a+5)^4]$$

from `[seq(p(i,a), i=1..10)]`.

$P(n, a) = a(a + n)^{n-1}$

First check the initial conditions: when $n = 1$, the number is a; when $n = 0$, the number is 1; when $a = 0$ and $n \geq 1$, the number is 0. By induction, only need to prove

$$a(a+n)^{n-1} = \sum_{k=0}^{n} \binom{n}{k} (a-1+k)(a+n-1)^{n-k-1}.$$

$P(n, a) = a(a+n)^{n-1}$

Proof.

$$f(x) := \sum_{k=0}^{n} \binom{n}{k} (a+k-1) x^{n-k-1}.$$

$$\begin{aligned}
f(x) &= \frac{a-1}{x} \sum_{k=0}^{n} \binom{n}{k} x^{n-k} + \sum_{k=0}^{n} k \binom{n}{k} x^{n-k-1} \\
&= \frac{a-1}{x} \sum_{k=0}^{n} \binom{n}{k} x^{n-k} + n \sum_{k=0}^{n} \binom{n}{k} x^{n-k-1} - \sum_{k=0}^{n} (n-k) \binom{n}{k} x^{n-k-1} \\
&= \frac{a-1+n}{x} \sum_{k=0}^{n} \binom{n}{k} x^{n-k} - \sum_{k=0}^{n} (n-k) \binom{n}{k} x^{n-k-1} \\
&= \frac{a-1+n}{x} (1+x)^n - n(1+x)^{n-1}.
\end{aligned}$$

$$P(n, a) = f(a+n-1) = a(a+n)^{n-1}.$$

Labelled rooted forests satisfy the same recurrence

We consider labelled rooted forests with a components where the roots are $1, 2, \ldots, a$ and the total number of vertices are $a+n$. Let $T(n,a)$ denote the number of such forests.

If $n=0$, $T(n,a)=1$. If $n \geq 1$ and $a=0$, $T(n,a)=0$.

Consider the number of neighbors of the vertex 1, remove them with their subtrees and delete vertex 1. Then there are $a+k-1$ components and $n-k$ non-root vertices. Hence

$$T(n,a) = \sum_{k=0}^{n} \binom{n}{k} T(n-k, a+k-1).$$

Bijection between a-parking functions and labelled rooted forests

Since their numbers are the same for the same n, of course there are lots of bijections.

We discover or possibly re-discover a bijection.

Bijection

The bijection can be best demonstrated by examples. Let's consider a 2-parking function of length 7: 5842121.

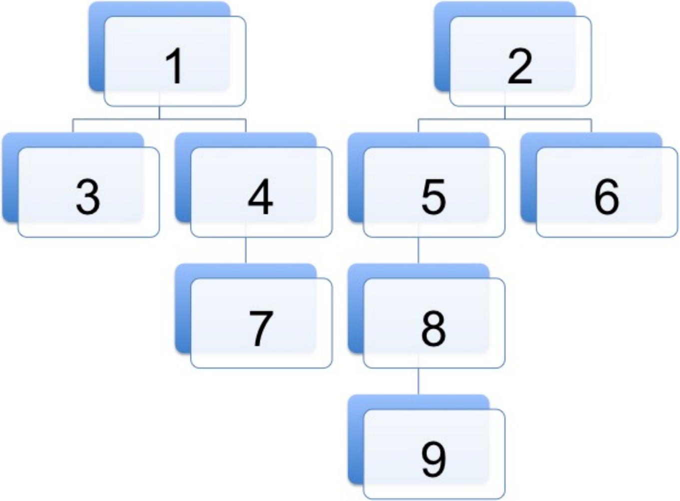
$$\begin{array}{rccccccc} vertices: & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 2-parkingfunction: & 5 & 8 & 4 & 2 & 1 & 2 & 1 \end{array}$$

Sort the second line:

$$\begin{array}{rccccccc} vertices: & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 2-parkingfunction: & 1 & 1 & 2 & 2 & 4 & 5 & 8 \end{array}$$

Bijection

Interpret the sorted version as follows: the parent of vertices 3 and 4 is 1, 5's and 6's parent is 2, etc. Hence we have the following forest.



Bijection

But we are not done yet, because this forest corresponds to the sorted version, not the original one. If when we sort the second line, the first line's elements move accordingly, then we will have

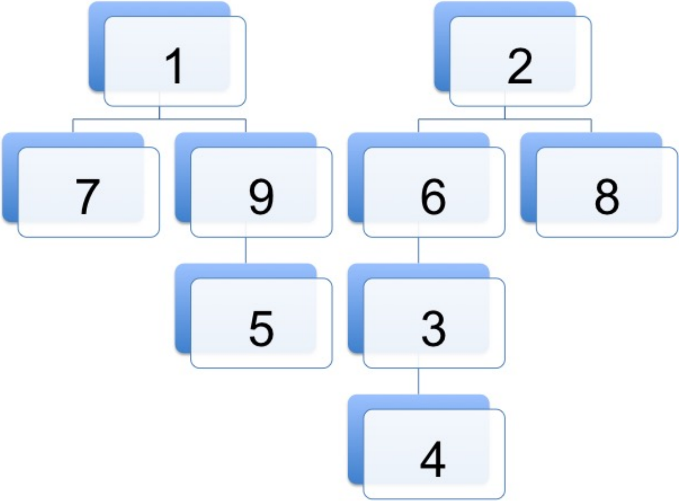
$$\begin{array}{rccccccc} vertices: & 7 & 9 & 6 & 8 & 5 & 3 & 4 \\ 2-parkingfunction: & 1 & 1 & 2 & 2 & 4 & 5 & 8 \end{array}$$

Compare the first line with that of the above sorted version, we have a map:

$$\begin{array}{ccccccc} 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 7 & 9 & 6 & 8 & 5 & 3 & 4 \end{array}$$

Bijection

So the 2-parking function 5842121 is mapped to the following forest:

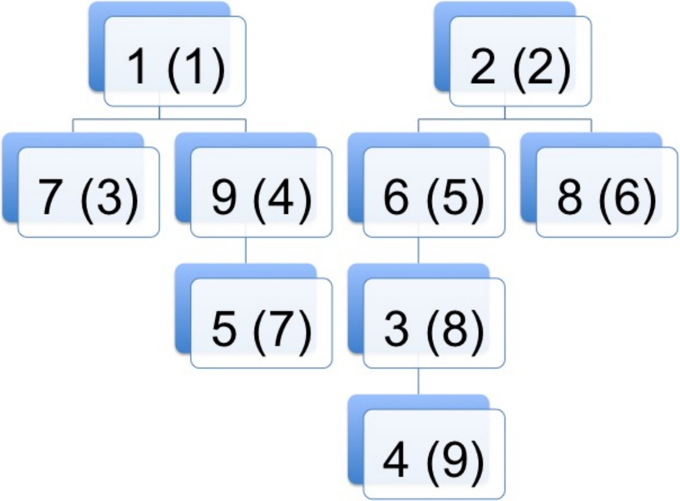


Bijection

One convention is that when we draw the forests, for the same parent, we always place its children in an increasing order (from left to right). Conversely, if we already have the above forest and we'd like to map it to a 2-parking function, then we start with indexing each vertex. The rule is we start from the first level, i.e. the root and start from the left, then we index the vertices 1, 2, . . . as follows with indexes in the bracket:

Bijection

After indexing, we have the forest:



Bijection

Now let the first line still be 3456789. For each of them in the first line, the second line number should be the index of its parent. Then we have

$$\begin{array}{rccccccc} vertices: & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 2-parkingfunction: & 5 & 8 & 4 & 2 & 1 & 2 & 1 \end{array}$$

From enumeration to statistics

Often in enumerative combinatorics, the class of interest has natural 'statistics', like height, weight, and IQ for humans, and one is interested rather than, for a finite set A,

$$|A| := \sum_{a \in A} 1,$$

called the *naive counting*, and getting a number (obviously a non-negative integer), by the so-called *weighted counting*,

$$|A|_x := \sum_{a \in A} x^{f(a)},$$

where $f := A \to Z$ is the statistic in question. To go from the weighted enumeration (a certain Laurent polynomial) to straight enumeration, one sets $x = 1$, i.e. $|A|_1 = |A|$.

From enumeration to statistics

The usual scenario is not just **one** specific set A, but a sequence of sets $\{A_n\}_{n=0}^{\infty}$, and then the enumeration problem is to have an efficient description of the numerical sequence $a_n := |A_n|$, ready to be looked-up (or submitted) to the OEIS, and its corresponding sequence of polynomials $P_n(x) := |A_n|_x$.

It often happens that the statistic f, defined on A_n, has a *scaled limiting distribution*. In other words, if you draw a *histogram* of f on A_n, and do the obvious *scaling*, they get closer and closer to a certain *continuous* curve, as n goes to infinity.

The scaling is as follows. Let $E_n(f)$ and $Var_n(f)$ the *expectation* and *variance* of the statistic f defined on A_n, and define the *scaled* random variable, for $a \in A_n$, by

$$X_n(a) := \frac{f(a) - E_n(f)}{\sqrt{Var_n(f)}}.$$

The sum and area statistics of a-parking functions

Let $\mathcal{P}(n,a)$ be the set of a-parking functions of length n.
A natural statistic is the sum

$$Sum(p_1,\dots,p_n) := p_1 + p_2 + \cdots + p_n = \sum_{i=1}^{n} p_i.$$

Another statistic is

$$Area(p) := \frac{n(2a+n-1)}{2} - Sum(p).$$

Let $P(n,a)(x)$ be the weighted analog of $P(n,a)$, according to Sum, i.e.

$$P(n,a)(x) \ := \sum_{p\in\mathcal{P}(n,a)} x^{Sum(p)}.$$

Analogously, let $Q(n,a)(x)$ be the weighted analog of $P(n,a)$, according to $Area$, i.e.

$$Q(n,a)(x) \ := \sum_{p\in\mathcal{P}(n,a)} x^{Area(p)}.$$

The sum and area statistics of a-parking functions

Clearly, one can easily go from one to the other

$$Q(n,a)(x) = x^{(2a+n-1)n/2} P(n,a)(x^{-1}),$$

$$P(n,a)(x) = x^{(2a+n-1)n/2} Q(n,a)(x^{-1}).$$

There are similar recurrence relations

$$P(n,a)(x) = x^n \sum_{k=0}^{n} \binom{n}{k} P(n-k, a+k-1)(x),$$

subject to the initial conditions $P(0,a)(x) = 1$ and $P(n,0)(x) = 0$. Equivalently,

$$Q(n,a)(x) = \sum_{k=0}^{n} \binom{n}{k} x^{k(k+2a-3)/2} Q(n-k, a+k-1)(x),$$

subject to the initial conditions $Q(0,a)(x) = 1$ and $Q(n,0)(x) = 0$.

Finding the expectation

The expectation of the sum statistic, $E_{sum}(n,a)$ is given by

$$E_{sum}(n,a) = \frac{P'(n,a)(1)}{P(n,a)(1)} = \frac{P'(n,a)(1)}{a(a+n)^{n-1}},$$

$$E_{sum}(n,a) = \frac{n(a+n+1)}{2} - \frac{1}{2}\sum_{j=1}^{n}\frac{n!}{(n-j)!(a+n)^{j-1}}.$$

$$E_{area}(n,a) = \frac{n\,(a-2)}{2} + \frac{1}{2}\sum_{j=1}^{n}\frac{n!}{(n-j)!(a+n)^{j-1}}.$$

$$E_{area}(n,1) = -\frac{n}{2} + \frac{1}{2}\sum_{j=1}^{n}\frac{n!}{(n-j)!(n+1)^{j-1}}.$$

A435

$$E_{area}(n,1) = -\frac{n}{2} + \frac{1}{2}W_{n+1},$$

where W_n is the **iconic** quantity,

$$W_n = \frac{n!}{n^{n-1}} \sum_{k=0}^{n-2} \frac{n^k}{k!},$$

proved by Riordan and Sloane to be the expectation of another very important quantity, the sum of the heights on labeled rooted trees on n vertices.

$$W_n \sim \sqrt{\pi/2}\, n^{\frac{3}{2}}.$$

In addition to its considerable mathematical interest, this quantity, W_n, has great *historical significance*, it was the *first sequence* , sequence A435 of the amazing On-Line Encyclopedia of Integer Sequences (OEIS), now with almost 300000 sequences!

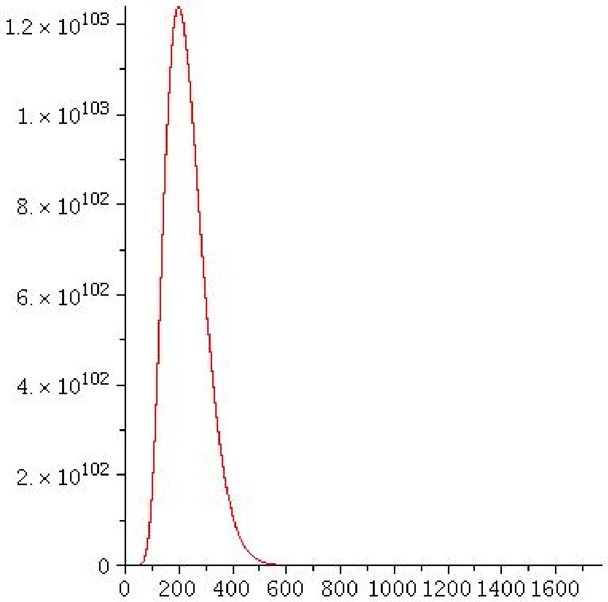
The limiting distribution

The limiting distribution of

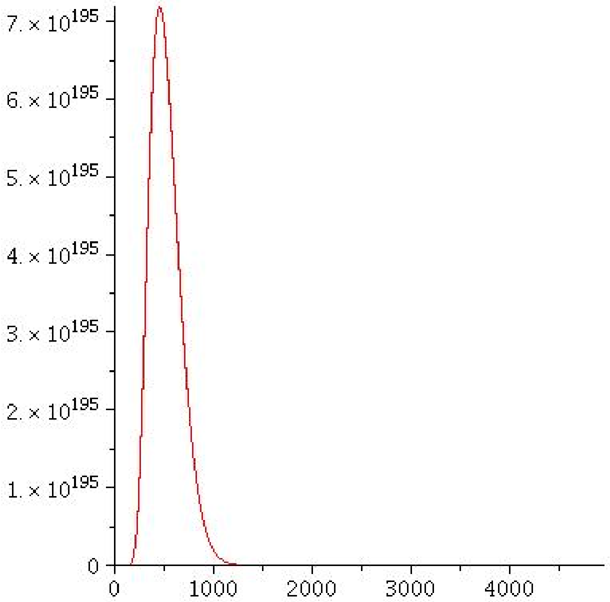
$$X_n(p) := \frac{Area(p) - E_n}{\sqrt{Var_n}}$$

is Airy distribution as proved by David Aldous, Svante Janson,and Chassaing and Marcket.

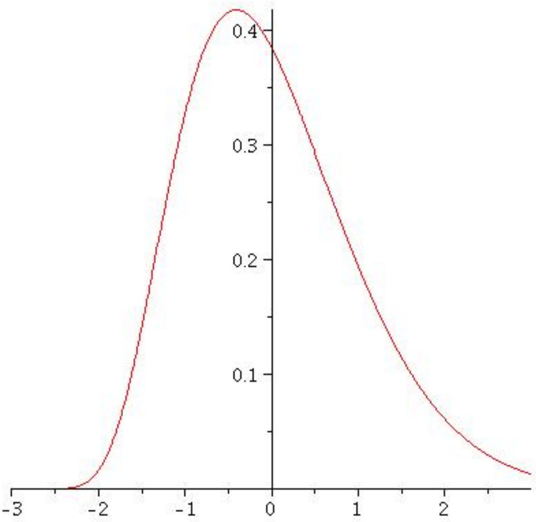
Histogram of the area of parking Functions of length 60



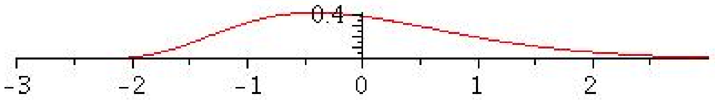
Histogram of the area of parking Functions of length 100



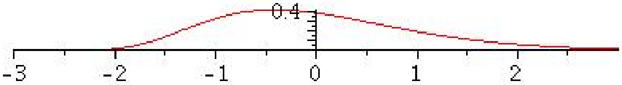
The scaled distribution of the area of parking functions of length 70



The scaled distribution of the area of parking functions of length 100

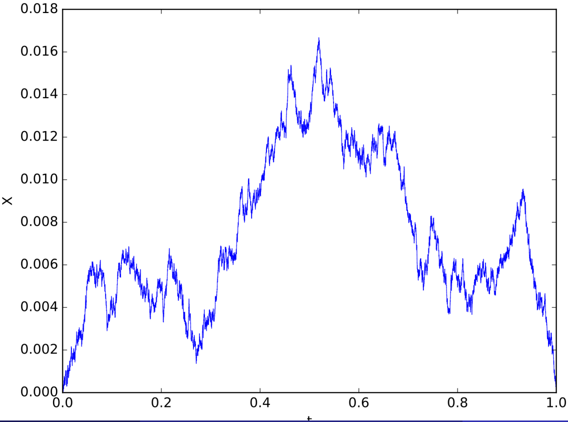


The scaled distribution of the area of parking functions of length 120



Airy Distribution

The Airy distribution function describes the probability distribution of the area under a Brownian excursion over a unit interval. Surprisingly, this function has appeared in a number of seemingly unrelated problems, mostly in computer science and graph theory.



Exact expression for the factorial moment

Let $E_1(n, a) := E_{area}(n, a)$ be the expectation of the area statistic on a-parking functions of length n, given above, and let $E_k(n, a)$ be the k-th factorial moment

$$E_k(n, a) \; := \; \frac{Q^{(k)}(n, a)(1)}{a(a+n)^{n-1}},$$

then there exist polynomials $A_k(n, a)$ and $B_k(n, a)$ such that

$$E_k(n, a) \; = A_k(n, a) \; + \; B_k(n, a) \, E_1(n, a).$$

The second factorial moment

The **second** factorial moment of the area statistic on parking functions of length n is

$$-\frac{7}{3}(n+1)\,E_1(n)+\frac{5}{12}\,n^3-\frac{1}{12}\,n^2-\frac{1}{3}\,n,$$

and asymptotically it equals $\frac{5}{12}\cdot n^3+O(n^{5/2})$.

The third factorial moment

The **third** factorial moment of the area statistic on parking functions of length n is

$$-\frac{175}{192}\,n^4-\frac{283}{192}\,n^3+\frac{199}{192}\,n^2+\frac{259}{192}\,n$$
$$+\left(\frac{15}{32}\,n^3+\frac{521}{96}\,n^2+\frac{1219}{96}\,n+\frac{743}{96}\right)\,E_1(n),$$

and asymptotically it equals $\frac{15}{128}\sqrt{2\pi}\cdot n^{9/2}+O(n^4)$.

The fourth factorial moment

The **fourth** factorial moment of the area statistic on parking functions of length n is

$$\frac{221}{1008}\,n^6+\frac{63737}{30240}\,n^5+\frac{101897}{15120}\,n^4+\frac{22217}{5040}\,n^3-\frac{1375}{189}\,n^2-\frac{187463}{30240}\,n$$

$$+\left(-\frac{35}{16}\,n^4-\frac{449}{27}\,n^3-\frac{130243}{2520}\,n^2-\frac{7409}{105}\,n-\frac{503803}{15120}\right)\,E_1(n),$$

and asymptotically it equals $\frac{221}{1008}\cdot n^6+O(n^{11/2})$.

The fifth factorial moment

The **fifth** factorial moment of the area statistic on parking functions of length n is

$$-\frac{105845}{110592}n^7-\frac{2170159}{290304}n^6-\frac{99955651}{3870720}n^5-\frac{30773609}{725760}n^4-\frac{94846903}{11612160}n^3+$$

$$\frac{24676991}{483840}n^2+\frac{392763901}{11612160}n$$

$$+\left(\frac{565}{2048}n^6+\frac{1005}{128}n^5+\frac{9832585}{165888}n^4+\frac{1111349}{5184}n^3+\frac{826358527}{1935360}n^2\right.$$

$$\left.+\frac{159943787}{362880}n+\frac{1024580441}{5806080}\right)E_1(n),$$

and asymptotically it equals $\frac{565}{8192}\sqrt{2\pi}\cdot n^{15/2}+O(n^7)$.

The sixth factorial moment

The **sixth** factorial moment of the area statistic of parking functions of length n is

$$\frac{82825}{576576}n^9+\frac{373340075}{110702592}n^8+\frac{9401544029}{332107776}n^7$$
$$+\frac{14473244813}{127733760}n^6+\frac{414139396709}{1660538880}n^5$$
$$+\frac{88215445651}{332107776}n^4-\frac{18783816473}{332107776}n^3-\frac{643359542029}{1660538880}n^2-\frac{358936540409}{1660538880}n$$
$$+\left(-\frac{3955}{2048}n^7-\frac{186349}{6144}n^6-\frac{259283273}{1161216}n^5-\frac{119912501}{129024}n^4-\frac{149860633081}{63866880}n^3\right.$$
$$\left.-\frac{601794266581}{166053888}n^2-\frac{864000570107}{276756480}n-\frac{921390308389}{830269440}\right)E_1(n),$$

and asymptotically it equals $\frac{82825}{576576}\cdot n^9+O(n^{17/2})$.

Gordian Knot of C-finite Ansatz

The Gordian Knot

Once upon a time there was a knot that no one could untangle, it was so complicated. Then came Alexander the Great and, in one second, cut it with his sword.

Methodology

In this part, we will describe two case studies where we know that the generating functions are rational, and it is easy to bound the degree of the denominator (alias the order of the recurrence satisfied by the sequence). Hence a simple-minded, empirical, approach of computing the first few terms and then 'fitting' a recurrence (equivalently rational function) is possible.

C-finite

A sequence of numbers $\{a(n)\}$ $(0 \leq n < \infty)$ is C-finite if it satisfies a *linear recurrence equation with constant coefficients*. For example the Fibonacci sequence that satisfies $F(n) - F(n-1) - F(n-2) = 0$ for $n \geq 2$.

Generating Function

A sequence $\{a(n)\}_{n=0}^{\infty}$ is C-finite if and only if its (ordinary) *generating function* $f(t) := \sum_{n=0}^{\infty} a(n)\, t^n$ is a **rational function** of t, i.e. $f(t) = P(t)/Q(t)$ for some *polynomials* $P(t)$ and $Q(t)$. For example, famously, the generating function of the Fibonacci sequence is $t/(1-t-t^2)$.

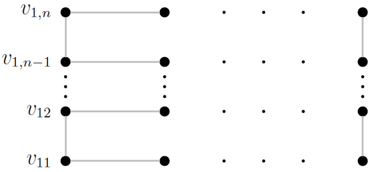
Grid Graph

Definition

The $k \times n$ grid graph $G_k(n)$ is the following graph given in terms of its vertex set V and edge set E:

$$V = \{v_{ij} | 1 \leq i \leq k, 1 \leq j \leq n\},$$

$$S = \{\{v_{ij}, v_{i'j'}\} || i - i'| + |j - j'| = 1\}.$$



A lexicographic ordering on $\mathcal{B}_k$

Let $\mathcal{B}_k$ be the collection of all set-partitions of $[k]$. A lexicographic ordering on $\mathcal{B}_k$ is defined as follows:
Given two partitions P_1 and P_2 of $[k]$, for $i \in [k]$, let X_i be the block of P_1 containing i and Y_i be the block of P_2 containing i. Let j be the minimum number such that $X_i \neq Y_i$. Then $P_1 < P_2$ iff
1. $|P_1| < |P_2|$ or
2. $|P_1| = |P_2|$ and $X_j \prec Y_j$ where $\prec$ denotes the normal lexicographic order.

For example, here is the ordering for $k = 3$:

$$\mathcal{B}_3 = \{\{\{1,2,3\}\},\{\{1\},\{2,3\},\{\{1,2\},\{3\}\},\{\{1,3\},\{2\}\},\{\{1\},\{2\},\{3\}\}\}$$

For simplicity, we can rewrite it as follows:

$$\mathcal{B}_3 = \{123, 1/23, 12/3, 13/2, 1/2/3\}.$$

Some Definitions

Definition

Given a spanning forest F of $G_k(n)$, the partition induced by F is obtained from the equivalence relation

$$i \sim j \iff v_{n,i}, v_{n,j} \text{ are in the same component of } F.$$

Definition

Given a spanning forest F of $G_k(n)$ and a set-partition P of $[k]$, we say that F is consistent with P if:

1. The number of trees in F is precisely $|P|$.
2. P is the partition induced by F.

Transfer Matrix

Let E_n be the set of edges $E(G_k(n)\backslash E(G_k(n-1))$, then E_n has $2k-1$ members.
Given a forest F of $G_k(n-1)$ and some subset $X \subseteq E_n$, we can combine them to get a forest of $G_k(n)$. We just need to know how many subsets of E_n can transfer a forest consistent with some partition to a forest consistent with another partition.

Transfer Matrix

Definition

Given two partitions P_1 and P_2 in $\mathcal{B}_k$, a subset $X \subseteq E_n$ transfers from P_1 to P_2 if a forest consistent with P_1 becomes a forest consistent with P_2 after the addition of X. In this case, we write $X \diamond P_1 = P_2$.
With the above definitions, it is natural to define a $B_k \times B_k$ transfer matrix A_k by the following:

$$A_k(i,j) = |\{A \subseteq E_{n+1} | A \diamond P_j = P_i\}|.$$

Examples

$$A_2 = \begin{bmatrix} 3 & 1 \\ 2 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 8 & 3 & 3 & 4 & 1 \\ 4 & 3 & 2 & 2 & 1 \\ 4 & 2 & 3 & 2 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 3 & 2 & 2 & 2 & 1 \end{bmatrix}$$

Examples

$$A_4 = \begin{bmatrix}
21 & 8 & 9 & 11 & 8 & 14 & 11 & 15 & 3 & 3 & 4 & 3 & 4 & 5 & 1 \\
9 & 8 & 6 & 4 & 4 & 6 & 5 & 8 & 3 & 3 & 4 & 2 & 2 & 2 & 1 \\
6 & 4 & 9 & 4 & 4 & 4 & 4 & 4 & 3 & 2 & 2 & 3 & 2 & 2 & 1 \\
3 & 0 & 0 & 3 & 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
9 & 4 & 6 & 5 & 8 & 6 & 4 & 8 & 2 & 3 & 2 & 3 & 4 & 2 & 1 \\
1 & 0 & 0 & 1 & 0 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
3 & 1 & 0 & 1 & 0 & 2 & 3 & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 4 & 6 & 4 & 3 & 4 & 3 & 4 & 3 & 2 & 2 & 2 & 2 & 2 & 1 \\
5 & 4 & 4 & 3 & 4 & 6 & 3 & 4 & 2 & 3 & 2 & 2 & 2 & 2 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
5 & 3 & 6 & 3 & 4 & 4 & 4 & 4 & 2 & 2 & 2 & 3 & 2 & 2 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
4 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 2 & 2 & 2 & 2 & 2 & 2 & 1
\end{bmatrix}$$

Thoughts

With the transfer matrices, the recurrence relation, sequence and generating function can follow immediately. However, with the famous theorem on next page, we are able to find initial data easily and guess a recurrence relation from the data.

The Matrix Tree Theorem

Theorem (The Matrix Tree Theorem)

If $A = (a_{ij})$ is the adjacency matrix of an arbitrary graph G, then the number of spanning trees is equal to the determinant of any co-factor of the Laplacian matrix L of G. Taking the (n, n) co-factor, we have that the number of spanning trees of G equals

$$\begin{vmatrix} a_{12} + \cdots + a_{1n} & -a_{12} & \dots & -a_{1,n-1} \\ -a_{21} & a_{21} + \cdots + a_{2n} & \dots & -a_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n-1,1} & -a_{n-1,2} & \dots & a_{n-1,1} + \cdots + a_{n-1,n} \end{vmatrix}.$$

GuessRec

GuessRec is a Maple procedure which accepts inputs of a list of numbers and outputs a conjectured linear recurrence relation.

```
> GuessRec1 := proc(L, d) local eq, var, a, i, n :
  if nops(L) ≤ 2*d + 2 then
   print(`The list must be of size >=`, 2*d + 3 ) :
   RETURN(FAIL) :
  fi:
  var := {seq(a[i], i = 1 ..d)} :
  eq := {seq(L[n]-add(a[i]*L[n-i], i = 1 ..d), n = d + 1 ..nops(L))} :
  var := solve(eq, var) :
  if var = NULL then
   RETURN(FAIL) :
  else
   RETURN([[op(1 ..d, L)], [seq(subs(var, a[i]), i = 1 ..d)]]) :
  fi:
  end:
```

```
> GuessRec := proc(L) local gu, d :

  for d from 1 to trunc(nops(L)/2)-2 do
   gu := GuessRec1(L, d) :
   if gu ≠ FAIL then
    RETURN(gu) :
   fi:
  od:
  FAIL :

  end:
```

CtoR

`CtoR` is a Maple procedure which accepts inputs of a recurrence relation and outputs a generating function.

```
> CtoR := proc(S, t) local D1, i, N1, L1, f, f1, L :
  if not (type(S, list) and nops(S) = 2 and type(S[1], list) and type(S[2], list)
       and nops(S[1]) = nops(S[2]) and type(t, symbol) ) then
    print(`Bad input`) :
    RETURN(FAIL) :
  fi:

  D1 := 1-add(S[2][i]*t^i, i = 1..nops(S[2])) :
  N1 := add(S[1][i]*t^(i-1), i = 1..nops(S[1])) :
  L1 := expand(D1*N1) :
  L1 := add(coeff(L1, t, i)*t^i, i = 0..nops(S[1])-1) :
  f := L1/D1 :
  L := degree(D1, t) + 10 :
  f1 := taylor(f, t = 0, L + 1) :
  if expand([seq(coeff(f1, t, i), i = 0..L)]) ≠ expand(SeqFromRec(S, L + 1)) then
  print([seq(coeff(f1, t, i), i = 0..L)], SeqFromRec(S, L + 1)) :
   RETURN(FAIL) :
  else
   RETURN(f) :
  fi:
  end:
```

General Idea

- Have a long enough list L ($|L| > d + 2$) of data of the number of spanning trees in the $k \times n$ grid graph $G_k(n)$ where k is fixed.
- Use `GuessRec` to guess a recurrence relation from the list of data.
- Use `CtoR` to find a generating function for the recurrence relation.
- With the generating function, it is easy to get the number of spanning trees in $G_k(n)$ even for very large n. This is much faster than calculating the determinant of the co-factor of the Laplacian matrix for large n.
- Since the number of states of a grid graph is finite for fixed k, the numbers of spanning trees for different n are a C-finite sequence. Its generating function must be a unique rational function. Hence our result is rigorous.

The generating function for $G_1(n)$

$$F_1 = \frac{t}{1-t}$$

The generating function for $G_2(n)$

$$F_2 = \frac{t}{t^2 - 4\,t + 1}$$

The generating function for $G_3(n)$

$$F_3 = \frac{-t^3 + t}{t^4 - 15\,t^3 + 32\,t^2 - 15\,t + 1}$$

The generating function for $G_4(n)$

$$F_4 = \frac{t^7 - 49\,t^5 + 112\,t^4 - 49\,t^3 + t}{t^8 - 56\,t^7 + 672\,t^6 - 2632\,t^5 + 4094\,t^4 - 2632\,t^3 + 672\,t^2 - 56\,t + 1}$$

The generating function for $G_5(n)$

$$\left(-t^{15} + 1440\,t^{13} - 26752\,t^{12} + 185889\,t^{11} - 574750\,t^{10} + 708928\,t^{9} - 708928\,t^{7} + 574750\,t^{6} - 185889\,t^{5} + 26752\,t^{4} - 1440\,t^{3} + t\right) \Big/ \left(t^{16} - 209\,t^{15} + 11936\,t^{14} - 274208\,t^{13} + 3112032\,t^{12} - 19456019\,t^{11} + 70651107\,t^{10} - 152325888\,t^{9} + 196664896\,t^{8} - 152325888\,t^{7} + 70651107\,t^{6} - 19456019\,t^{5} + 3112032\,t^{4} - 274208\,t^{3} + 11936\,t^{2} - 209\,t + 1\right)$$

The generating function for $G_6(n)$

$$\begin{aligned}
&\big(t^{31} - 33359\,t^{29} + 3642600\,t^{28} - 173371343\,t^{27} + 4540320720\,t^{26} - 70164186331\,t^{25} + 634164906960\,t^{24} - 2844883304348\,t^{23} - 1842793012320\,t^{22} + 104844096982372\,t^{21} - 678752492380560\,t^{20} \\
&\quad + 2471590551535210\,t^{19} - 5926092273213840\,t^{18} + 9869538714631398\,t^{17} - 11674018886109840\,t^{16} + 9869538714631398\,t^{15} - 5926092273213840\,t^{14} + 2471590551535210\,t^{13} \\
&\quad - 678752492380560\,t^{12} + 104844096982372\,t^{11} - 1842793012320\,t^{10} - 2844883304348\,t^{9} + 634164906960\,t^{8} - 70164186331\,t^{7} + 4540320720\,t^{6} - 173371343\,t^{5} + 3642600\,t^{4} - 33359\,t^{3} + t\big) \Big/ \\
&\quad \big(t^{32} - 780\,t^{31} + 194881\,t^{30} - 22377420\,t^{29} + 1419219792\,t^{28} - 55284715980\,t^{27} + 1410775106597\,t^{26} - 24574215822780\,t^{25} + 300429297446885\,t^{24} - 2629946465331120\,t^{23} \\
&\quad + 16741727755133760\,t^{22} - 78475174345180080\,t^{21} + 273689714665707178\,t^{20} - 716370537293731320\,t^{19} + 1417056251105102122\,t^{18} - 2129255507292156360\,t^{17} \\
&\quad + 2437932520099475424\,t^{16} - 2129255507292156360\,t^{15} + 1417056251105102122\,t^{14} - 716370537293731320\,t^{13} + 273689714665707178\,t^{12} - 78475174345180080\,t^{11} \\
&\quad + 16741727755133760\,t^{10} - 2629946465331120\,t^{9} + 300429297446885\,t^{8} - 24574215822780\,t^{7} + 1410775106597\,t^{6} - 55284715980\,t^{5} + 1419219792\,t^{4} - 22377420\,t^{3} + 194881\,t^{2} - 780\,t + 1\big)
\end{aligned}$$

The generating function for $G_7(n)$

$$
\begin{aligned}
&(-t^{47} - 142\,t^{46} + 661245\,t^{45} - 279917500\,t^{44} + 53184503243\,t^{43} - 5570891154842\,t^{42} + 341638600598298\,t^{41} - 11886702497030032\,t^{40} + 164458937576610742\,t^{39} + 437115847049245l828\,t^{38} \\
&- 2887373449568553013 42\,t^{37} + 7736513993329973661368\,t^{36} - 13158233876832285395 6994\,t^{35} + 1573202877300834187134466\,t^{34} - 13805721749199518460916737\,t^{33} \\
&+ 90975567796174070740787232\,t^{32} - 455915282590547643587452175\,t^{31} + 1747901867578637315747826286\,t^{30} - 5126323837327170557921412877\,t^{29} \\
&+ 11416779122947828869806142972\,t^{28} - 18924703166237080216745900796\,t^{27} + 22194247945745188489023284104\,t^{26} - 15563815847174688069871470516\,t^{25} \\
&+ 15563815847174688069871470516\,t^{23} - 22194247945745188489023284104\,t^{22} + 18924703166237080216745900796\,t^{21} - 11416779122947828869806142972\,t^{20} \\
&+ 5126323837327170557921412877\,t^{19} - 1747901867578637315747826286\,t^{18} + 455915282590547643587452175\,t^{17} - 90975567796174070740787232\,t^{16} + 13805721749199518460916737\,t^{15} \\
&- 1573202877300834187134466\,t^{14} + 131582338768322853956994\,t^{13} - 7736513993329973661368\,t^{12} + 288737344956855301342\,t^{11} - 4371158470492451828\,t^{10} - 164458937576610742\,t^{9} \\
&+ 11886702497030032\,t^{8} - 341638600598298\,t^{7} + 5570891154842\,t^{6} - 53184503243\,t^{5} + 279917500\,t^{4} - 661245\,t^{3} + 142\,t^{2} + t) \Big/ (t^{48} - 2769\,t^{47} + 2630641\,t^{46} - 1195782497\,t^{45} \\
&+ 305993127089\,t^{44} - 48551559344145\,t^{43} + 5083730101530753\,t^{42} - 366971376492201338\,t^{41} + 18871718211768417242\,t^{40} - 709234610141846974874\,t^{39} + 19874722637854592209338\,t^{38} \\
&- 422023241997789381263002\,t^{37} + 6880098547452856483997402\,t^{36} - 87057778313447181201990522\,t^{35} + 862879164715733847737203343\,t^{34} - 6750900711491569851736413311\,t^{33} \\
&+ 41958615314622858303912597215\,t^{32} - 208258356862493902206466194607\,t^{31} + 828959040281722890327985220255\,t^{30} - 2654944041424536277948746010303\,t^{29} \\
&+ 6859440538554030239641036025103\,t^{28} - 14324708604336971207868317957868\,t^{27} + 24214587194571650834572683444012\,t^{26} - 33166490975387358866518005011884\,t^{25} \\
&+ 36830850383375837481096026357868\,t^{24} - 33166490975387358866518005011884\,t^{23} + 24214587194571650834572683444012\,t^{22} - 14324708604336971207868317957868\,t^{21} \\
&+ 6859440538554030239641036025103\,t^{20} - 2654944041424536277948746010303\,t^{19} + 828959040281722890327985220255\,t^{18} - 208258356862493902206466194607\,t^{17} \\
&+ 41958615314622858303912597215\,t^{16} - 6750900711491569851736413311\,t^{15} + 862879164715733847737203343\,t^{14} - 87057778313447181201990522\,t^{13} + 6880098547452856483997402\,t^{12} \\
&- 422023241997789381263002\,t^{11} + 19874722637854592209338\,t^{10} - 709234610141846974874\,t^{9} + 18871718211768417242\,t^{8} - 366971376492201338\,t^{7} + 5083730101530753\,t^{6} \\
&- 48551559344145\,t^{5} + 305993127089\,t^{4} - 1195782497\,t^{3} + 2630641\,t^{2} - 2769\,t + 1)
\end{aligned}
$$

Remarks

Generally, for an arbitrary graph G, we consider the number of spanning trees in $G \times P_n$. With the same methodology, a list of data can be obtained empirically with which a generating function follows.

The statistic of the number of vertical edges

Let $ver(T)$ = the number of vertical edges in the spanning tree T, define the weight $w(T) = v^{ver(T)}$, then the weighted counting follows:

$$Ver_{k,n}(v) = \sum_{T \in \mathcal{F}_{k,n}} w(T)$$

where $\mathcal{F}_{k,n}$ is the set of spanning trees of $G_k(n)$.
Define the bivariate generating function

$$g_k(v,t) = \sum_{n=0}^{\infty} Ver_{k,n} t^n.$$

General Idea

The main tool for computing `VerGF` is still the Matrix Tree Theorem and `GuessRec`. But we need to modify the Laplacian matrix for the graph. Instead of letting $a_{ij} = -1$ for $i \neq j$ and $\{i, j\} \in E(G \times P_n)$, we should consider whether the edge $\{i, j\}$ is a vertical edge. If so, we let $a_{i,j} = -v, a_{j,i} = -v$. The diagonal elements which are $(-1)\times$ (the sum of the rest entries on the same row) should change accordingly.

The bivariate generating function for the weighted counting

$$g_2(v,t) = \frac{vt}{1-(2\,v+2)\,t+t^2}$$

$$g_3(v,t) = \frac{-t^3v^2+v^2t}{1-(3\,v^2+8\,v+4)\,t-(-10\,v^2-16\,v-6)\,t^2-(3\,v^2+8\,v+4)\,t^3+t^4}$$

The rest formulas are too long to display here.

Almost-diagonal matrices

So far, we have seen applications of the C-finite ansatz methodology for automatically computing generating functions for enumerating spanning trees/forests for certain infinite families of graphs.
The second case study is completely different, and in a sense more general, since the former framework may be subsumed in this new context.

Almost-diagonal matrices

Definition

Diagonal matrices A are square matrices in which the entries outside the main diagonal are 0, i.e. $a_{ij} = 0$ if $i \neq j$.

Definition

An almost-diagonal matrix A is a square matrices in which $a_{i,j} = 0$ if $j - i \geq k_1$ or $i - j \geq k_2$ for some fixed positive integers k_1, k_2 and $\forall i_1, j_1, i_2, j_2$, if $i_1 - j_1 = i_2 - j_2$, then $a_{i_1 j_1} = a_{i_2 j_2}$.

For simplicity, we use the notation $L =$[n, [the first k_1 entries in the first row], [the first k_2 entries in the first column]] to denote the $n \times n$ matrix with these specifications. Note that this notation already contains all information we need to reconstruct this matrix. For example, [6, [1,2,3], [1,4]] is the matrix

$$\begin{bmatrix} 1 & 2 & 3 & 0 & 0 & 0 \\ 4 & 1 & 2 & 3 & 0 & 0 \\ 0 & 4 & 1 & 2 & 3 & 0 \\ 0 & 0 & 4 & 1 & 2 & 3 \\ 0 & 0 & 0 & 4 & 1 & 2 \\ 0 & 0 & 0 & 0 & 4 & 1 \end{bmatrix}.$$

GFfamilyDet

Here is the Maple procedure `GFfamilyDet` which takes inputs (i) A: a name of a Maple procedure that inputs an integer n and outputs an $n \times n$ matrix according to some rule, e.g., the almost-diagonal matrices, (ii) a variable name t, (iii) two integers m and n which are the lower and upper bounds of the sequence of determinants we consider. It outputs a rational function in t, say $R(t)$, which is the generating function of the sequence.

```
> GFfamilyDet := proc(A, t, m, n) local i, rec, GF, B, gu, Denom, L, Numer :
  L := [seq(det(A(i)), i = 1 ..n)] :
  rec := GuessRec([op(m ..n, L)])[2] :
  gu := solve(B - 1 - add(t^i * rec[i] * B, i = 1 ..nops(rec)), {B}) :
  Denom := denom(subs(gu, B)) :
  Numer := Denom * (1 + add(L[i] * t^i, i = 1 ..n)) :
  Numer := add(coeff(Numer, t, i) * t^i, i = 0 ..degree(Denom, t)) :
  Numer / Denom :
  end:
```

Example

Similarly we have procedure `GFfamilyPer` for the permanent. Let's look at an example. `SampleB` is a sample procedure which outputs the $n \times n$ almost-diagonal matrix which the first row is $[2, 3]$ and the first column is $[2, 4, 5]$.

Then `GFfamilyDet(SampleB, t, 10, 50)` will return the generating function

$$-\frac{1}{45\,t^3 - 12\,t^2 + 2\,t - 1}.$$

Symbolic dynamic programming approach

Recall from Linear Algebra 101, the
Cofactor Expansion Let $|A|$ denote the determinant of an $n \times n$ matrix A, then

$$|A| = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} M_{ij}, \quad \forall i \in [n],$$

where M_{ij} is the (i, j)-minor.
We'd like to consider the Cofactor Expansion for almost-diagonal matrices along the first row. For simplicity, we assume while $a_{i,j} = 0$ if $j - i \geq k_1$ or $i - j \geq k_2$ for some fixed positive integers k_1, k_2, and if $-k_2 < j_1 - i_1 < j_2 - i_2 < k_1$, then $a_{i_1 j_1} \neq a_{i_2 j_2}$. Under this assumption, for any minors we obtain through recursive Cofactor Expansion along the first row, the dimension, the first row and the first column should provide enough information to reconstruct the matrix.

General Idea

- For an almost-diagonal matrix represented by $L =$[Dimension, [the first k_1 entries in the first row], [the first k_2 entries in the first column]], any minor can be represented by [Dimension, [entries in the first row up to the last nonzero entry], [entries in the first column up to the last nonzero entry]].
- Use `ExpandMatrixL` to do a one-step cofactor expansion along the first row.
- Use `ChildrenMatrixL` to find all "children" of an almost-diagonal matrix.
- After all "children" are found, we have a scheme S. By the cofactor expansion of any element in the scheme, a system of algebraic equation follows.
- Use `GFMatrixL` to solve the system of equations to get the generating function.

ExpandMatrixL

```
> ExpandMatrixL := proc(L, L1) local n, R, C, dim, R1, C1, i, r, S, candidate, newrow, newcol, gu, mu, temp, p, q, j :
  n := L[1] : R := L[2] : C := L[3] : p := nops(R) - 1 : q := nops(C) - 1 :
  dim := L1[1] :
  R1 := L1[2] :
  C1 := L1[3] :
  if R1 = [ ] or C1 = [ ] then
   return { } :
  elif R[1] ≠ C[1] or R1[1] ≠ C1[1] or dim > n then
   return fail :
  else
   S := { } :
   gu := [0$(n-1-q), seq(C[q-i+1], i = 0 ..q-1), op(R), 0$(n-1-p)] :
   candidate := [0$nops(R1), R1[-1]] :
   for i from 1 to nops(R1) do
    mu := R1[i] :
    for j from n-q to nops(gu) do
     if gu[j] = mu then
      candidate[i] := gu[j-1] :
     fi:
    od:
   od:
   for i from n-q to nops(gu) do
    if gu[i] = R1[2] then
     temp := i :
     break:
    fi:
   od:
   for i from 1 to nops(R1) do
    if i = 1 then
     mu := [R1[i]*(-1)^(i+1), [dim-1, [op(i+1 ..nops(candidate), candidate)], [seq(gu[temp-i], i = 1 ..temp-n+q)]]] :
     S := S union {mu} :
    else
     mu := [R1[i]*(-1)^(i+1), [dim-1, [op(1 ..i-1, candidate), op(i+1 ..nops(candidate), candidate)], [op(2 ..nops(C1), C1)]]] :
     S := S union {mu} :
    fi:
   od:
   return S :
```

ChildrenMatrixL

```
> ChildrenMatrixL := proc(L) local S, t, T, dim, U, u, s :
  dim := L[1]:
  S := {[op(2..3, L)]} :
  T := {seq([op(2..3, t[2])], t in ExpandMatrixL(L, L))} :
  while T minus S ≠ { } do
   U := T minus S :
   S := S union T :
   T := { } :
   for u in U do
    T := T union {seq([op(2..3, t[2])], t in ExpandMatrixL(L, [dim, op(u)]))} :
   od:
  od:
  for s in S do
   if s[1] = [ ] or s[2] = [ ] then
    S := S minus {s} :
   fi:
  od:
  S :
  end:
```

GFMatrixL

```
> GFMatrixL := proc(L, t) local S, dim, var, eq, n, A, i, result, gu, mu :
  dim := L[1]:
  S := ChildrenMatrixL(L) :
  S := [[op(2..3, L)], op(S minus {[op(2..3, L)]})]:
  n := nops(S) :
  var := {seq(A[i], i = 1..n)} :
  eq := { }:
  for i from 1 to 1 do
   result := ExpandMatrixL(L, [dim, op(S[i])]) :
   for gu in result do
   if gu[2][2] = [ ] or gu[2][3] = [ ] then
    result := result minus {gu} :
   fi:
   od:
   eq := eq union {A[i] - 1 - add(gu[1]*t*A[CountRank(S, [op(2..3, gu[2])])], gu in result)} :
  od:
  for i from 2 to n do
   result := ExpandMatrixL(L, [dim, op(S[i])]) :
   for gu in result do
   if gu[2][2] = [ ] or gu[2][3] = [ ] then
    result := result minus {gu} :
   fi:
   od:
   eq := eq union {A[i] - add(gu[1]*t*A[CountRank(S, [op(2..3, gu[2])])], gu in result)} :
  od:
  gu := solve(eq, var)[1]:
  subs(gu, A[1]) :
```

Example

`GFMatrixL([20, [2, 3], [2, 4, 5]], t)` returns

$$-\frac{1}{45\,t^3 - 12\,t^2 + 2\,t - 1}.$$

Analysis of Quicksort Algorithms

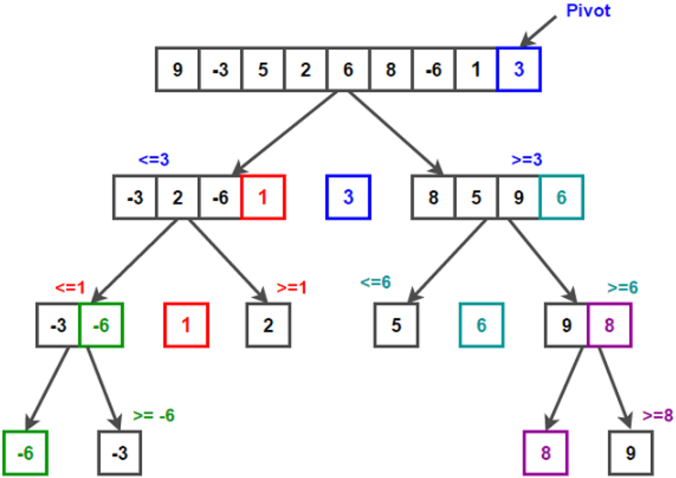
Sorting Algorithms

Sorting algorithms are very important in computer science and technology industry. There are many different sorting algorithms.

- Quicksort: average complexity $O(n \log n)$
- Merge sort: average complexity $O(n \log n)$
- Bubble sort: average complexity $O(n^2)$
- Insertion sort: average complexity $O(n^2)$
- Selection sort: average complexity $O(n^2)$

Quicksort

Quicksort is the most widely used sorting algorithm.



Human Approach

- Little research has been focusing on the explicit formula of the performance and higher moments of Quicksort algorithms.
- It seems that only the explicit formulas of the expectation and variance are previously known via very complicated human approach.
- Higher moments seem to be inaccessible via human approach.

The number of comparisons

The number of comparisons of Quicksort is independent of the implementation. Let C_n be the random variable: the number of comparisons and $c_n = E(C_n)$, then

$$c_n = \frac{1}{n}\sum_{k=1}^{n}((n-1) + c_{k-1} + c_{n-k}) = (n-1) + \frac{1}{n}\sum_{k=1}^{n}(c_{k-1} + c_{n-k})$$

$$= (n-1) + \frac{2}{n}\sum_{k=1}^{n} c_{k-1}.$$

Methodology

Let

$$H_k(n) := \sum_{i=1}^{n} \frac{1}{i^k}.$$

Our educated guess is that the moments should be a multivariate polynomial involving n, $H_1(n), \ldots, H_m(n)$ for some m. m depends on the order of the moment.

Expectation

Theorem

$$E[C_n] = 2(n+1)H_1(n) - 4n.$$

Variance

Theorem (Knuth)

$$var[C_n] = n(7\,n + 13)\ -\ 2\,(n+1)\,H_1(n) - 4\,(n+1)^2 H_2(n).$$

The third moment

Theorem (Zeilberger)

The third moment about the mean of C_n is

$$-n(19\,n^2+81\,n+104)+H_1(n)(14\,n+14)+12\,(n+1)^2H_2(n)+16\,(n+1)^3H_3(n).$$

The fourth moment

Theorem (Zeilberger)

The fourth moment about the mean of C_n is

$$\frac{1}{9}\,n(2260\,n^3+9658\,n^2+15497\,n+11357)-2\,(n+1)(42\,n^2+78\,n+77)H_1(n)$$
$$+12\,(n+1)^2(H_1(n))^2+(-4\,(42\,n^2+78\,n+31)(n+1)^2+48\,(n+1)^3H_1(n))H_2(n)$$
$$+48\,(n+1)^4(H_2(n))^2-96\,(n+1)^3H_3(n)-96\,(n+1)^4H_4(n).$$

The number of swaps

The number of swaps is much more complicated than the number of comparisons since it is dependent on the specific variant. It might be also more significant since a swap usually takes more computing resources than a comparison.

- Variant Nulla
- Variant I
- Variant II
- Variant III
- Variant IV
- Variant V

Variant Nulla

In Variant Nulla, we have the original list L. Every time when a pivot is chosen and the comparisons are done, we will have two new lists L_1 and L_2 where elements in L_1 are less than the pivot and elements in L_2 are greater than the pivot. So there is really no swap involved in this variant. But it is not space-efficient. It is also not time-efficient because we need to generate so many new lists and merge them.

Variant I

- Choose the first (or equivalently, the last) element in the list of length n as the pivot, then we compare the other elements with the pivot.
- We compare the second element with the pivot first. If it is greater than the pivot, it stays where it is, otherwise we insert it before the pivot.
- Though this is somewhat different from the “traditional swap,” we define this operation as a swap.
- Generally, every time we find an element smaller than the pivot, we insert it before the pivot.

Variant I

Let $P_n(t)$ be the probability generating function for the number of swaps X_n, i.e.,

$$P_n(t) = \sum_{k=0}^{\infty} P(X_n = k)\, t^k,$$

where for only finitely many integers k, we have that $P(X_n = k)$ is nonzero.

We have the recurrence relation

$$P_n(t) = \frac{1}{n} \sum_{k=1}^{n} P_{k-1}(t) P_{n-k}(t) t^{k-1},$$

with the initial condition $P_0(t) = P_1(t) = 1$ because for any fixed $k \in \{1, 2, \ldots, n\}$, the probability that the pivot is the k-th smallest is $\frac{1}{n}$ and there are exactly $k-1$ swaps when the pivot is the k-th smallest.

Variant I

Theorem

The expectation of the number of swaps of Quicksort for a list of length n under Variant I is

$$E[X_n] = (n+1)H_1(n) - 2n.$$

Variant I

Theorem

The variance of X_n is

$$2n(n+2) - (n+1)H_1(n) - (n+1)^2 H_2(n).$$

Variant I

Theorem

The third moment about the mean of X_n is

$$-\frac{9}{4}n(n+3)^2 + (4n+4)H_1(n) + 3(n+1)^2 H_2(n) + 2(n+1)^3 H_3(n).$$

Theorem

The fourth moment about the mean of X_n is

$$\frac{1}{18}n(335n^3+1568n^2+3067n+2770)-3(n+1)(4n^2+8n+9)H_1(n)$$
$$+3(n+1)^2H_1(n)^2+(-(12n^2+24n+19)(n+1)^2+6(n+1)^3H_1(n))H_2(n)$$
$$+3(n+1)^4H_2(n)^2-12(n+1)^3H_3(n)-6(n+1)^4H_4(n).$$

Variant II

- The second variant is similar to the first one. One tiny difference is that instead of choosing the first or last element as the pivot, the index of the pivot is chosen uniformly at random.
- For example, we choose the i-th element, which is the k-th smallest, as the pivot. Then we compare those non-pivot elements with the pivot.
- If $i \neq 1$, the first element will be compared with the pivot first. If it is smaller than the pivot, it stays there, otherwise it is moved to the end of the list.
- After comparing all the left-side elements with the pivot, we look at those elements whose indexes are originally greater than i. If they are greater than the pivot, no swap occurs; otherwise insert them before the pivot.

Variant II

- Let $Q(n, k, i, t)$ be the probability generating function of the number of swap in the first partition step when the length of the list is n, the pivot is the i-th element and is the k-th smallest.
- The number of swaps equals to the number of elements which are before k and greater than k or after k and smaller than k.
- If there are j elements which are before k and smaller than k, the number of swaps is $i + k - 2 - 2j$.
- The range of j is $[\max(k - 1 - n + i, 0), \min(i - 1, k - 1)]$.

Variant II

$$Q(n,k,i,t) =$$

$$\sum_{j=\max(k-1-n+i,0)}^{\min(i-1,k-1)} \binom{i-1}{j} \prod_{s=0}^{j-1} \frac{k-1-s}{n-1-s} \prod_{s=0}^{i-j-2} \frac{n-k-s}{n-1-j-s} t^{i+k-2-2j},$$

$$P_n(t) = \frac{1}{n^2} \sum_{k=1}^{n} \sum_{i=1}^{n} P_{k-1}(t) P_{n-k}(t) Q(n,k,i,t),$$

Variant II

Theorem

The expectation of the number of swaps of Quicksort for a list of length n under Variant II is

$$E[X_n] = (n+1)H_1(n) - 2n.$$

Theorem

The variance of X_n is

$$\frac{1}{6}n(11n+17) - \frac{1}{3}(n+1)H_1(n) - (n+1)^2 H_2(n).$$

Variant II

Theorem

The third moment about the mean of X_n is

$$-\frac{1}{6}n(14n^2+57n+73)+(2n+2)H_1(n)+(n+1)^2H_2(n)+2(n+1)^3H_3(n).$$

Variant II

Theorem

The fourth moment about the mean of X_n is

$$\frac{1}{90}n(1496n^3+5531n^2+8527n+6922)-\frac{1}{15}(n+1)(55n^2+85n+173)H_1(n)$$

$$+\frac{1}{3}(n+1)^2H_1(n)^2$$

$$(-\frac{1}{3}(33n^2+51n+25)(n+1)^2+2(n+1)^3H_1(n))H_2(n)+3(n+1)^4H_2(n)^2$$

$$-4(n+1)^3H_3(n)-6(n+1)^4H_4(n).$$

Variant III

The third variant is the most used version: the in-place Quicksort.

```
def partition(arr,low,high):
    i = ( low-1 )
    pivot = arr[high]
    for j in range(low , high):
        if  arr[j] <= pivot:
            i = i+1
            arr[i],arr[j] = arr[j],arr[i]
    arr[i+1],arr[high] = arr[high],arr[i+1]
    return ( i+1 )


def quickSort(arr,low,high):
    if low < high:
        pi = partition(arr,low,high)
        quickSort(arr, low, pi-1)
        quickSort(arr, pi+1, high)
```

Variant III

$$P_n(t) = \frac{1}{n}\sum_{k=1}^{n} P_{k-1}(t)P_{n-k}(t)t^k$$

Theorem

The expectation of the number of swaps of Quicksort for a list of length n under Variant III

$$E[X_n] = (n+1)H_1(n) - \frac{4}{3}n - \frac{1}{3}.$$

Theorem

The variance of the number of swaps of Quicksort for a list of length n under Variant III

$$var[X_n] = 2n^2 + \frac{187}{45}n + \frac{7}{45} - \frac{2}{3n} - (n^2+2n+1)H_2(n) - (n+1)H_1(n).$$

Variant IV

In Variant III, every time when `A[j]` $\leq$ `pivot`, we swap `A[i]` with `A[j]`. However, it is a waste to swap them when $i = j$. If we modify the algorithm such that a swap is performed only when the indexes $i \neq j$, the expected cost will be reduced.

Variant IV

Lemma

Let $Y_n(k)$ be the number of swaps needed in the first partition step in an in-place Quicksort without swapping the same index for a list L of length n when the pivot is the k-th smallest element, then

$$Y_n(k) = \begin{cases} |\{i \in [n] \mid L[i] \leq pivot \land \exists j < i, L[j] > pivot\}| & k < n \\ 0 & k = n \end{cases}.$$

Variant IV

When $k < n$, the probability that there are s swaps is

$$\binom{k-1}{k-s}\frac{(k-s)!(n-k)(n-k-2+s)!}{(n-1)!} = \frac{n-k}{n-1}\frac{\binom{k-1}{k-s}}{\binom{n-2}{k-s}}.$$

Therefore the probability generating function

$$Q(n,k,t) = \sum_{s=1}^{k}\frac{n-k}{n-1}\frac{\binom{k-1}{k-s}}{\binom{n-2}{k-s}}t^s.$$

$$P_n(t) = \frac{1}{n}\sum_{k=1}^{n} P_{k-1}(t)P_{n-k}(t)Q(n,k,t)$$

Variant IV

Theorem

The expectation of the number of swaps of Quicksort for a list of length n under Variant IV

$$E[X_n] = (n+2)H_1(n) - \frac{5}{2}n - \frac{1}{2}.$$

Theorem

The variance of the number of swaps of Quicksort for a list of length n under Variant IV

$$var[X_n] = 2n^2 - \frac{215}{12}n + \frac{1}{12} + (11n+14)H_1(n) - (n^2-2n-2)H_2(n)$$
$$-(2n+2)H_1(n)^2$$

Variant V

- This variant might not be practical, but we find that it is interesting as a combinatorial model.
- As is well-known, if a close-to-median element is chosen as a pivot, the Quicksort algorithm will have better performance than average in this case. Hence if additional information is available so that the probability distribution of chosen pivots is no longer a uniform distribution but something Gaussian-like, it is to our advantage.
- Assume that the list is a permutation of $[n]$ and we are trying to sort it, pretending that we do not know the sorted list must be $[1, 2, \dots, n]$. Now the rule is that we choose the first and last number in the list, look at the numbers and choose the one which is closer to the median. If the two numbers are equally close to the median, then choose one at random.

Variant V

Considering symmetry, $Pr^{(n)}(pivot = k) = Pr^{(n)}(pivot = n + 1 - k)$, so we only need to consider $1 \le k \le (n+1)/2$. When n is even, let $n = 2m$. Then $Pr^{(n)}(pivot = k) = \frac{4k-3}{(2m-1)2m}$. When n is odd, let $n = 2m - 1$, then $Pr^{(n)}(pivot = k) = \frac{4k-3}{(2m-1)(2m-2)}$ when $k < m$ and $Pr^{(n)}(pivot = m) = \frac{2}{2m-1}$.
With this minor modification, the recurrence relation for $P_n(t)$ follows.

$$P_n(t) = \sum_{k=1}^{n} P_{k-1}(t)P_{n-k}(t)Q(n,k,t)Pr^{(n)}(pivot = k)$$

with the initial condition $P_0(t) = P_1(t) = 1$.

We obtain the following recurrence relation for the expectation of the number of swaps:

$$
\begin{aligned}
&\frac{n\,(n-1)\,(n+1)\,\left(124300966127\,n^2+675726017124\,n+650712665709\right)}{(n+7)\,(n+6)\,(n+5)\,\left(17482518431\,n^2+498259584530\,n+1781699167235\right)}-\frac{2\,n\,(n+2)\,(n+1)\,\left(216417203729\,n^2+919675870477\,n+92300503754\right)N}{(n+7)\,(n+6)\,(n+5)\,\left(17482518431\,n^2+498259584530\,n+1781699167235\right)}\\
&+\frac{(n+2)\,(n+1)\,\left(350982431977\,n^3+911174904749\,n^2-6154705279825\,n-16672650937797\right)N^2}{(n+7)\,(n+6)\,(n+5)\,\left(17482518431\,n^2+498259584530\,n+1781699167235\right)}\\
&+\frac{4\,(n+2)\,\left(70891742279\,n^4+1238063051796\,n^3+6234404912517\,n^2+8329928119664\,n-5599544780136\right)N^3}{(n+7)\,(n+6)\,(n+5)\,\left(17482518431\,n^2+498259584530\,n+1781699167235\right)}\\
&-\frac{(n+3)\,\left(492765916535\,n^4+5467106459287\,n^3+8863038802013\,n^2-71364745940767\,n-204534459680620\right)N^4}{(n+7)\,(n+6)\,(n+5)\,\left(17482518431\,n^2+498259584530\,n+1781699167235\right)}\\
&+\frac{2\,(n+4)\,\left(74633719171\,n^3-190797245354\,n^2-6602055805627\,n-18802118680054\right)N^5}{(n+7)\,(n+5)\,\left(17482518431\,n^2+498259584530\,n+1781699167235\right)}+N^6
\end{aligned}
$$

Dual-pivot Quicksort

The probability generating function $P_n(t)$ of the total number of comparisons C_n of dual-pivot Quicksort is

$$P_n(t) = \frac{1}{\binom{n}{2}} \sum_{j=2}^{n} \sum_{i=1}^{j-1} P_{i-1}(t) P_{j-i-1}(t) P_{n-j}(t) t^{2n-i-2}$$

with the initial condition $P_0(t) = P_1(t) = 1$ and $P_2(t) = t$.

Dual-pivot Quicksort

Theorem

The expectation of the number of comparisons in dual-pivot Quicksort algorithms is

$$E[C_n] = 2(n+1)H_1(n) - 4n.$$

Theorem

The variance of the number of comparisons in dual-pivot Quicksort algorithms is

$$var[C_n] = n(7\,n+13) \; - \; 2\,(n+1)\,H_1(n) - 4\,(n+1)^2 H_2(n).$$

Theorem

The third moment about the mean of C_n is

$$-n(19\,n^2+81\,n+104)+H_1(n)(14\,n+14)+12\,(n+1)^2 H_2(n)+16\,(n+1)^3 H_3(n).$$

Dual-pivot Quicksort

Theorem

The fourth moment about the mean of C_n is

$$\frac{1}{9}\, n(2260\, n^3 + 9658\, n^2 + 15497\, n + 11357) - 2\,(n+1)(42\, n^2 + 78\, n + 77) H_1(n)$$
$$+12\,(n+1)^2 (H_1(n))^2 + (-4\,(42\, n^2 + 78\, n + 31)(n+1)^2 + 48\,(n+1)^3 H_1(n)) H_2(n)$$
$$+48\,(n+1)^4 (H_2(n))^2 - 96\,(n+1)^3 H_3(n) - 96\,(n+1)^4 H_4(n).$$

Observation: They are exactly the same with the 1-pivot Quicksort!

Dual-pivot Quicksort

How about the number of swaps?
As a toy model, we do an analogue of Variant I. The first and last elements are chosen as the pivot. Let's say they are i and j. If $i > j$ then we swap them and still call the smaller pivot i. For each element less than i, we move it to the left of i, and for each element greater than j, we move it to the right of j and call this kind of operations a swap.

Dual-pivot Quicksort

$$P_n(t) = \frac{1}{\binom{n}{2}}(\frac{1}{2} + \frac{1}{2}t)\sum_{j=2}^{n}\sum_{i=1}^{j-1} P_{i-1}(t)P_{j-i-1}(t)P_{n-j}(t)t^{n-1+i-j}$$

with the initial conditions $P_0(t) = P_1(t) = 1$ and $P_2(t) = \frac{1}{2} + \frac{1}{2}t$.

Theorem

The expectation of the number of swaps in the above dual-pivot Quicksort variant is

$$E[X_n] = \frac{4}{5}(n+1)H_1(n) - \frac{39}{25}n - \frac{1}{100}.$$

Three-pivot Quicksort

- How to sort the pivots? 1-pivot Quicksort.
- How to partition the list? Binary search.
- How to sort a list or sublist containing less than three elements? 1-pivot Quicksort.

Three-pivot Quicksort

The recurrence relation for the probability generating function $P_n(t)$ of the total number of comparisons for 3-pivot Quicksort of a list of length n is

$$P_n(t) =$$

$$\frac{1}{\binom{n}{3}} \sum_{k=3}^{n} \sum_{j=2}^{k-1} \sum_{i=1}^{j-1} P_{i-1}(t) P_{j-i-1}(t) P_{k-j-1}(t) P_{n-k}(t) \left(\frac{2}{3} t^{2n-3} + \frac{1}{3} t^{2n-4}\right)$$

with initial conditions $P_0(t) = P_1(t) = 1, P_2(t) = t$ and $P_3(t) = \frac{2}{3} t^3 + \frac{1}{3} t^2$.

Three-pivot Quicksort

Theorem

The expected number of comparisons C_n of 3-pivot Quicksort for a list of length n satisfies the following recurrence relation:

$$C_{n+4} = \frac{(n+1)(12n+7)}{(n+4)(3n+1)} C_{n+3} - 3\frac{(n+1)(6n+5)n}{(n+4)(n+3)(3n+1)} C_{n+2}$$

$$+ \frac{(12n^4+13n^3-12n^2+59n+24)}{(3n+1)(n+4)(n+3)(n+2)} C_{n+1} - \frac{(3n+4)(n^2-5n+12)}{(n+4)(n+3)(3n+1)} C_n.$$

k-pivot Quicksort and remarks

- Generally for a long enough list (the length $n \to \infty$), the more pivots the better.
- For a real-world application, the best strategy would be that we adjust the number of pivots when the length of its sublists varies.
- For 1-pivot Quicksort, it might be interesting to see whether there is any sampling method for the pivot which can significantly improve the efficiency.

Peaceable Queens Problem

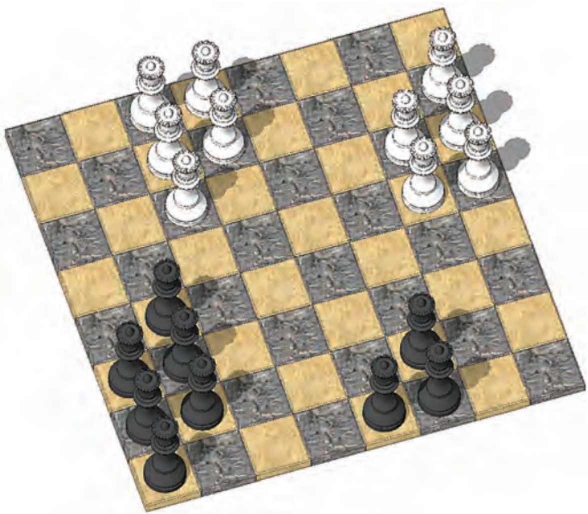
What is the peaceable queens problem?

What is the maximal number, m, such that it is possible to place m white queens and m black queens on an $n \times n$ chess board, so that no queen attacks a queen of the opposite color?

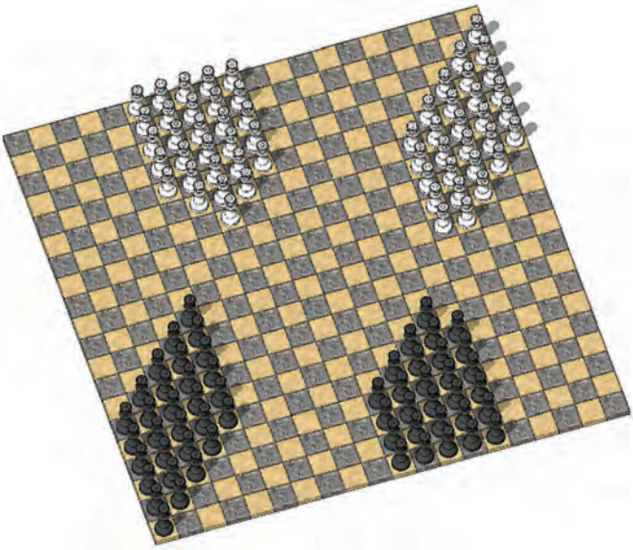
One of three solutions on 5 × 5 board

The following nice pictures are from https://www.ams.org/journals/notices/201809/rnoti-p1062.pdf By Dr. Neil Sloane.

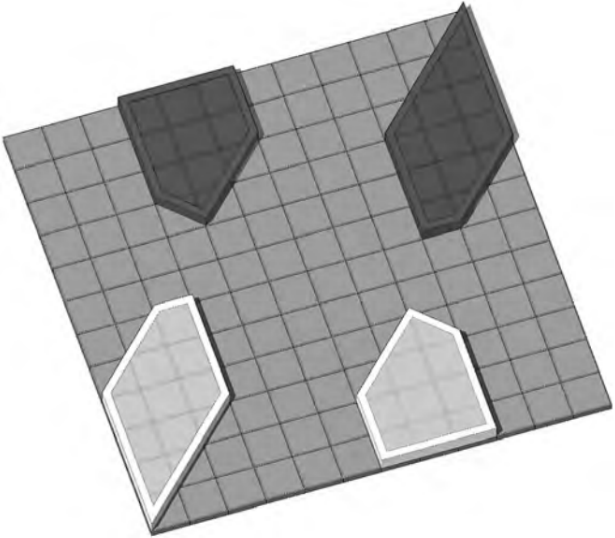
A solution on 8 × 8 board



A conjectured solution on 20 × 20 board



A general construction by Jubin



Known terms in the sequence

Currently only fifteen terms are known:

n:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$a(n)$:	0	0	1	2	4	5	7	9	12	14	17	21	24	28	32

Strategy

Consider this peaceable queens problem as a continuous problem by normalizing the chess board to be the unit square $U := [0,1]^2 = \{(x,y) \mid 0 \leq x, y \leq 1\}$. Let $W \subseteq U$ be the region where white queens are located. Then the non-attacking region B of W can be defined as

$$B = \{(x,y) \in U \mid \forall (u,v) \in W, x \neq u, y \neq v, x+y \neq u+v, y-x \neq v-u\}.$$

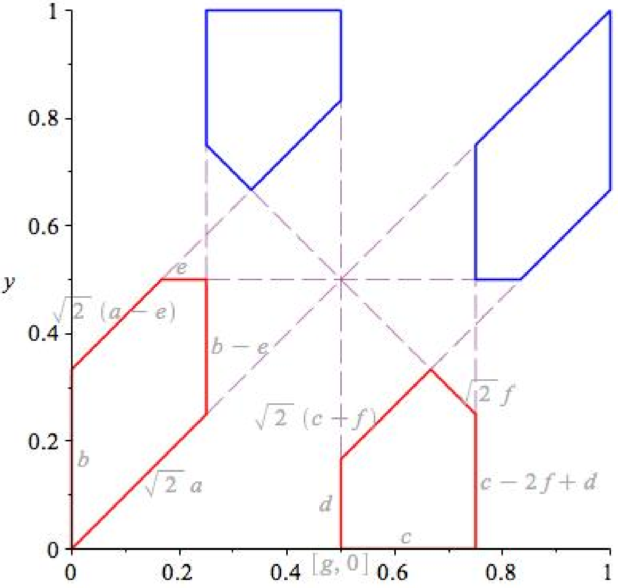
So the continuous version of the peaceable queens problem is to find

$$\max_{W \in Borel(U)} (\min(\mathrm{Area}(W), \mathrm{Area}(B))).$$

Observation

- The queen is able to move any number of squares vertically, horizontally and diagonally
- W should be a convex polygon or a disjoint union of convex polygons whose boundary consists of vertical, horizontal and slope ± 1 line segments.
- Otherwise we can increase the area of white queens without decreasing the area of black queens.
- Each component is at most an octagon.

Jubin's construction



We would like to prove this is at least a local optimum. By assuming that the white queens are placed in the union of the interiors of the following two pentagons

$$[\,[0,0]\,,\,[a,a]\,,\,[a,a+b-e]\,,\,[a-e,a+b-e]\,,\,[0,b]\,],$$

and

$$[\,[g,0]\,,\,[g+c,0]\,,\,[g+c,c-2\,f+d]\,,\,[g+c-f,c-f+d]\,,\,[g,d]\,],$$

it's not hard to find the region of black queens, and the areas of them. Then by setting the two areas equal and using Lagrange multiplier, we can find all the extreme points.

Jubin's construction

The white queens are located inside the pentagons

$$[\,[0,0]\,,\,[\frac{1}{4},\frac{1}{4}]\,,\,[\frac{1}{4},\frac{1}{2}]\,,\,[\frac{1}{6},\frac{1}{2}]\,,\,[0,\frac{1}{3}]\,],$$

and

$$[\,[\frac{1}{2},0]\,,\,[\frac{3}{4},0]\,,\,[\frac{3}{4},\frac{1}{4}]\,,\,[\frac{2}{3},\frac{1}{3}]\,,\,[\frac{1}{2},\frac{1}{6}]\,].$$

Jubin's construction

The black queens reside inside the pentagons

$$[\,[\frac{1}{2},1]\,,\,[\frac{1}{4},1]\,,\,[\frac{1}{4},\frac{3}{4}]\,,\,[\frac{1}{3},\frac{2}{3}]\,,\,[\frac{1}{2},\frac{5}{6}]\,],$$

and

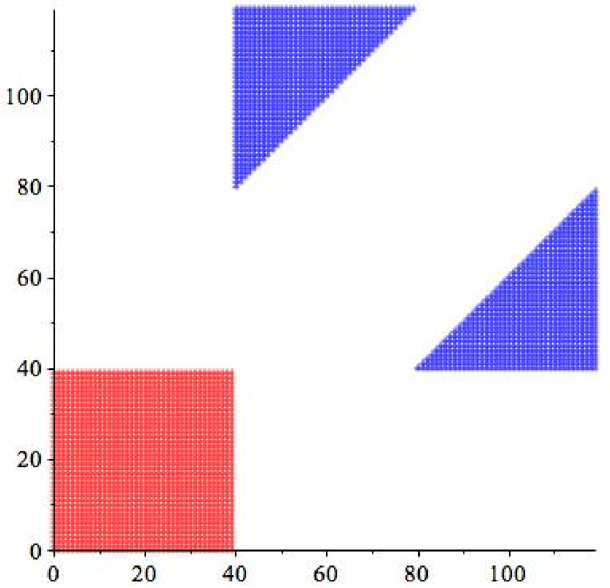
$$[\,[1,1]\,,\;\;[\frac{3}{4},\frac{3}{4}]\,,\,[\frac{3}{4},\frac{1}{2}]\,,\,[\frac{5}{6},\frac{1}{2}]\,,\,[1,\frac{2}{3}]\,].$$

Jubin's construction

The maximum area of this configuration is

$$\frac{7}{48}.$$

A single rectangle on 120 × 120 board



A single rectangle

Let the rectangle for white queens be $[[0,0], [a,0], [a,b], [0,b]]$. We'd like to find the maximum of ab under the condition

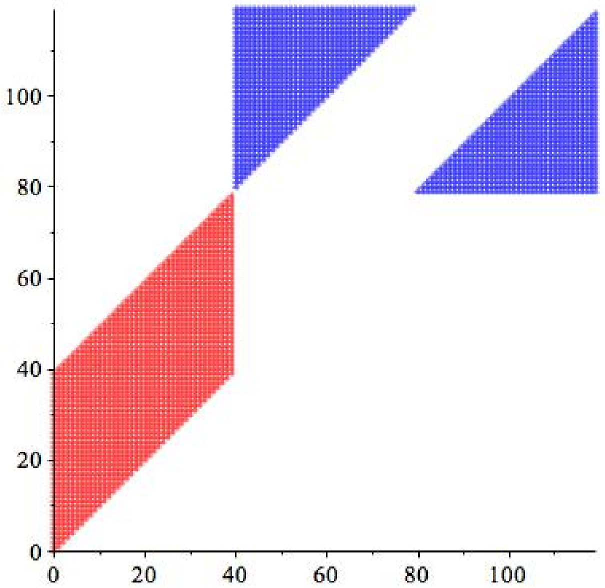
$$ab = (\max(1-a-b,0))^2, \quad 0 \le a, b \le 1.$$

We get

$$a = b = \frac{1}{3}$$

and the largest area for peaceable queens when the configuration for white queens is a rectangle is $\frac{1}{9}$.

A single parallelogram on 120 × 120 board



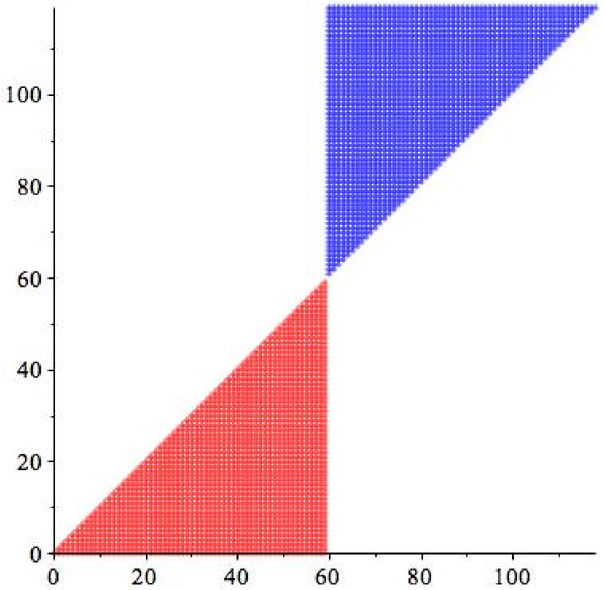
A single parallelogram

Let the parallelogram for white queens be $[[0,0], [a,a], [a,a+b], [0,b]]$. In this case the formulas of the areas of white and black queens are exactly the same as the previous case. Hence the when

$$a = b = \frac{1}{3}$$

we have the maximum area $\frac{1}{9}$.

A single triangle on 120 × 120 board



A single triangle

Let the white queens reside in the region $[[0,0],[0,a],[a,a]]$, then its area is

$$\frac{1}{2}a^2.$$

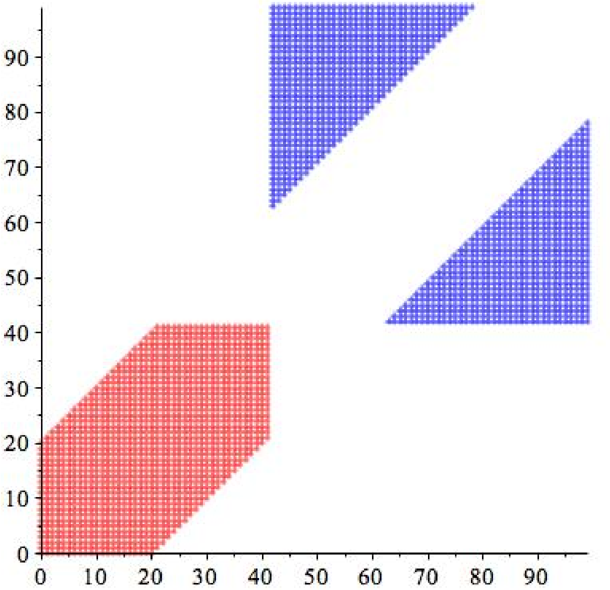
The area of black queens is $\frac{1}{2}(1-a)^2$. When $a=\frac{1}{2}$, the area is maximized at $\frac{1}{8}$.

The largest area for a single component

What could the largest area of white queens be when they have only one component?

- It is at most an octagon.
- It should be placed at some corner, e.g. lower left corner.
- Then it is at most a hexagon.
- Let's find out the maximum when it is a hexagon.

A single hexagon on 100 × 100 board



A single hexagon

The general shape is a hexagon

$$[\,[0,0]\,,\,[a,0]\,,\,[a+b,b]\,,\,[a+b,b+c]\,,\,[d,b+c]\,,\,[0,b+c-d]\,]$$

with four parameters. Then the area for white queens is

$$(a+b)(b+c)-\frac{1}{2}(b^2+d^2),$$

and the area for black queens is

$$\frac{1}{2}(1-a-b-c)^2+\frac{1}{2}(1-a-2b-c+d)^2.$$

A single hexagon

One of the local maximums found using Lagrange multipliers is when

$$a = c = d = \frac{1}{2},\ b = 0.$$

However, actually this is the optimal triangle with an area of $\frac{1}{8}$.

A single hexagon

Another local maximum is when $a = b = c = d$. In that case, we have

$$3a^2 = (1 - 3a)^2.$$

Hence when

$$a = \frac{3 - \sqrt{3}}{6} \approx 0.2113248654,$$

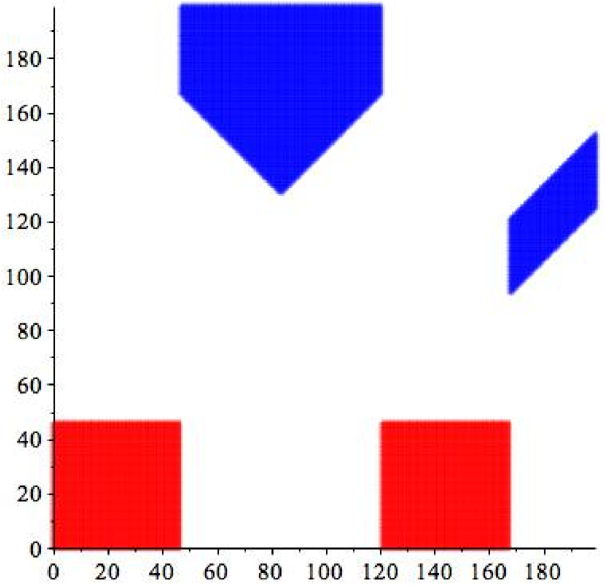
the area of white queens is maximized at

$$3a^2 = \frac{2 - \sqrt{3}}{2} \approx 0.1339745962.$$

Cylindrical algebraic decomposition

- Given a set S of polynomials in $\mathbb{R}^n$, a cylindrical algebraic decomposition is a decomposition of $\mathbb{R}^n$ into semi-algebraic connected sets called cells, on which each polynomial has constant sign, either $+$, - or 0. With such a decomposition it is easy to give a solution of a system of inequalities and equations defined by the polynomials, i.e. a real polynomial system.
- The cylindrical algebraic decomposition algorithm in quantifier elimination is applied to find out the exact optimal parameters and the maximum areas.

Two identical squares on 200 × 200 board



Two identical squares

The two squares are

$$[\,[0,0]\,,\ [a,0]\,,\ [a,a]\,,\ [0,a]\,]$$

and

$$[\,[s,0]\,,\ [s+a,0]\,,\ [s+a,a]\,,\ [s,a]\,].$$

Based on this configuration, the domain is

$$0 \le a \le \frac{1}{2}, \quad a \le s \le 1-a.$$

The area of white queens is

$$2a^2.$$

Two identical squares

Actually the formula for black queens is very complicated, especially when a is small there may be a lot of components for B. However, by experimentation (procedure `FindM2Square`), we found that for all mid-range $s \in [0.24, 0.76]$, a around 0.23 will always maximize the area. Then we just need to focus on the shape of B when a is not far from its optimum.

The area of black queens is

$$(s-a)(1-s-a)+\frac{1}{4}(s-a)^2+(\max(1-s-2a,0))^2+\max(s-2a,0)(1-s-a).$$

Two identical squares

The domain for a and s is a triangle. The area formula for black queens shows that the two lines $s = 2a$ and $s = 1 - 2a$ separate the domain into 4 regions. In each region, we have a polynomial formula for the area of black queens. Since the area of white queens W is just a simple formula of a, we need to maximize a with the condition $W = B$.

Two identical squares

When $s \geq 2a$ and $s \geq 1-2a$, by cylindrical algebraic decomposition we obtained

$$\begin{cases} \frac{1}{2}(-1+\sqrt{2}) \leq a < \frac{1}{27}(1+2\sqrt{7}) & s = \frac{4+a}{7} + \frac{2}{7}\sqrt{4-19a+9a^2} \\ \frac{1}{27}(1+2\sqrt{7}) \leq a < \frac{1}{18}(19-\sqrt{217}) & s = \frac{4+a}{7} \pm \frac{2}{7}\sqrt{4-19a+9a^2} \\ a = \frac{1}{18}(19-\sqrt{217}) & s = \frac{4+a}{7} - \frac{2}{7}\sqrt{4-19a+9a^2} \end{cases}.$$

When $s \leq 2a$ and $s \geq 1-2a$, the result is an empty set.
When $s \leq 2a$ and $s \leq 1-2a$, we obtained

$$\frac{2}{9} \leq a \leq \frac{1}{7}(3-\sqrt{2}), \quad s = 2-7a-2\sqrt{-2a+9a^2}.$$

When $s \geq 2a$ and $s \leq 1-2a$, we obtained

$$\frac{2}{9} \leq a \leq \frac{1}{27}(1+2\sqrt{7}), \quad s = 3a - \frac{2}{\sqrt{3}}\sqrt{1-7a+12a^2}.$$

Two identical squares

Comparing the four cases, we found that the largest area occurred in case 1, when

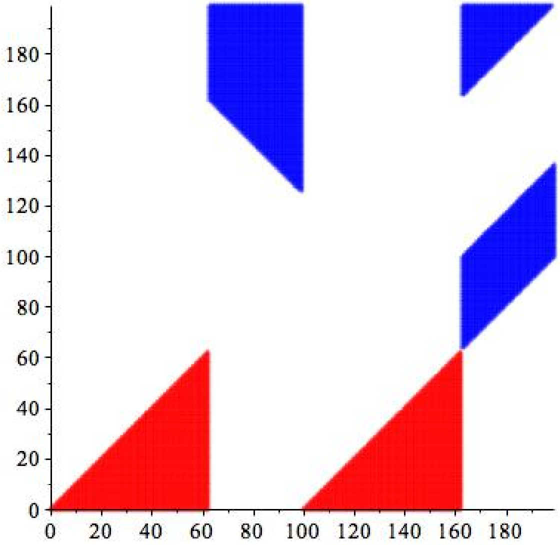
$$a = \frac{1}{18}(19 - \sqrt{217}) \approx 0.2371711193,$$

$$s = \frac{13}{18} - \frac{1}{126}\sqrt{217} \approx 0.6053101598.$$

The largest area is $\frac{289}{81} - \frac{19\sqrt{217}}{81} \approx 0.112500281$.

Two identical isosceles right triangles on 200 × 200 board

With the same orientation



Two identical isosceles right triangles with the same orientation

Similarly, also By CAD, when $s \geq 1 - 2a$

$$\begin{cases} \frac{1}{2}(2-\sqrt{2}) \leq a < \frac{1}{4}(-1+\sqrt{5}) & s = \frac{1}{2} + \frac{1}{2}\sqrt{3 - 12a + 8a^2} \\ \frac{1}{4}(-1+\sqrt{5}) \leq a < \frac{1}{4}(3-\sqrt{3}) & s = \frac{1}{2} \pm \frac{1}{2}\sqrt{3 - 12a + 8a^2} \\ a = \frac{1}{4}(3-\sqrt{3}) & s = \frac{1}{2} - \frac{1}{2}\sqrt{3 - 12a + 8a^2} \end{cases}.$$

When $s \leq 1 - 2a$, we obtained

$$\frac{1}{11}(5-\sqrt{3}) \leq a \leq \frac{1}{4}(-1+\sqrt{5}), s = 2a - \sqrt{2 - 10a + 12a^2}.$$

Hence the area is maximized when

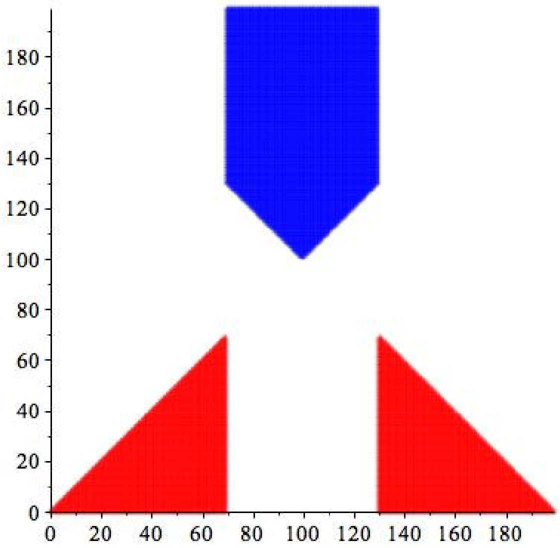
$$a = \frac{1}{4}(3-\sqrt{3}) \approx 0.316987298,$$

$$s = \frac{1}{2}.$$

The largest area is $\frac{3}{4} - \frac{3}{8}\sqrt{3} \approx 0.1004809470$.

Two identical isosceles right triangles on 200 × 200 board

With the opposite orientations



Two identical isosceles right triangles with the opposite orientations

If we take the two triangles to be

$$[\,[0,0]\,,\,[a,0]\,,\,[a,a]\,]$$

and

$$[\,[1-a,0]\,,\,[1,0]\,,\,[1-a,a]\,],$$

then the area of black queens is

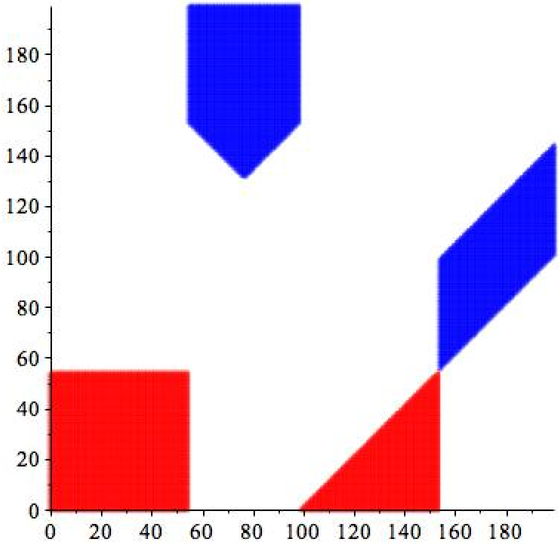
$$a(1-2a)+(\frac{1}{2}-a)^2=-a^2+\frac{1}{4}.$$

Equalizing the areas of white queens and black queens, we get

$$\mathrm{Area}(W)=a^2=\frac{1}{8},$$

which is greater than the optimal case of two identical isosceles right triangles with the same orientation.

One square and one triangle with the same side length on 200×200 board



One square and one triangle with the same side length

It is obtained that when $s \geq 1 - 2a$

$$\begin{cases} \frac{1}{2}(-2+\sqrt{6}) \leq a < \frac{1}{21}(1+\sqrt{22}) & s = \frac{4-a}{7} + \frac{1}{7}\sqrt{16-64a+22a^2} \\ \frac{1}{21}(1+\sqrt{22}) \leq a < \frac{2}{11}(8-\sqrt{42}) & s = \frac{4-a}{7} \pm \frac{1}{7}\sqrt{16-64a+22a^2} \\ a = \frac{2}{11}(8-\sqrt{42}) & s = \frac{4-a}{7} - \frac{1}{7}\sqrt{16-64a+22a^2} \end{cases},$$

and when $s \leq 1 - 2a$

$$\frac{1}{15}(6-\sqrt{6}) \leq a \leq \frac{1}{21}(1+\sqrt{22}), \quad s = \frac{7a}{3} - \frac{1}{3}\sqrt{12-72a+106a^2}.$$

Consequently, we have the maximized area when

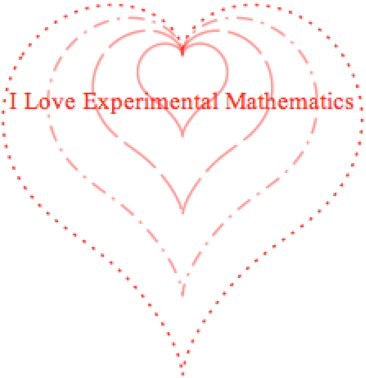
$$a = \frac{2}{11}(8-\sqrt{42}) \approx 0.276228965,$$

$$s = \frac{112}{33} - \frac{14}{33}\sqrt{42} - \frac{50}{33}\sqrt{7} + \frac{52}{33}\sqrt{6} \approx 0.495622162.$$

The largest area is $\frac{636}{121} - \frac{96}{121}\sqrt{42} \approx 0.1144536616.$

Summary

Summary



"I Love Experimental Mathematics", from π-day homework of Experimental Mathematics class in Spring 2018.

Thanks for Your Attention!