Permutations sortable by deques and by two stacks in parallel.

Andrew Elvey Price
Joint work with Tony Guttmann

The University of Melbourne

June 29, 2017

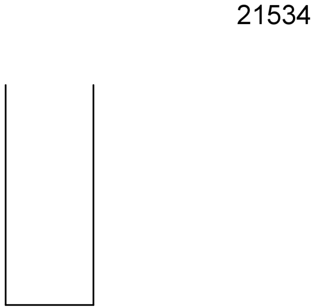
STACK SORTING

Stack Sorting

Stack sorting is a method of sorting a permutation where we are only allowed to use two operations: Put the next element from the input onto the stack, and output the top element of the stack.

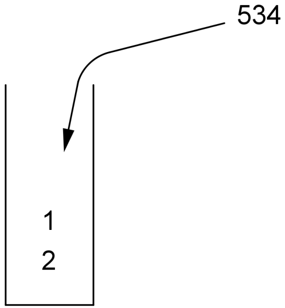
Stack Sorting

Stack sorting is a method of sorting a permutation where we are only allowed to use two operations: Put the next element from the input onto the stack, and output the top element of the stack.



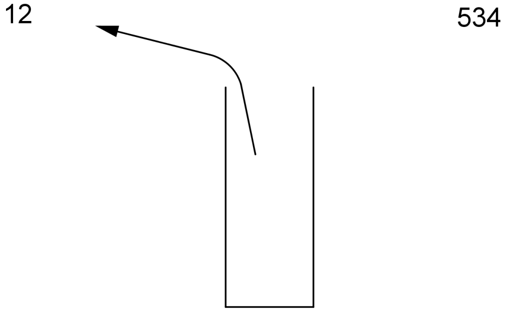
Stack Sorting

Stack sorting is a method of sorting a permutation where we are only allowed to use two operations: Put the next element from the input onto the stack, and output the top element of the stack.



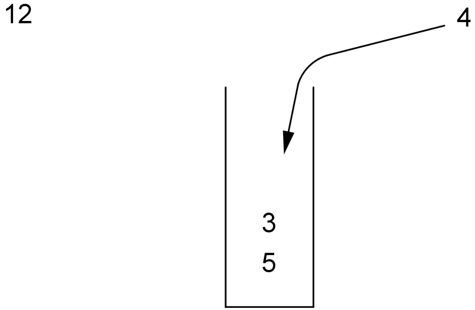
Stack Sorting

Stack sorting is a method of sorting a permutation where we are only allowed to use two operations: Put the next element from the input onto the stack, and output the top element of the stack.



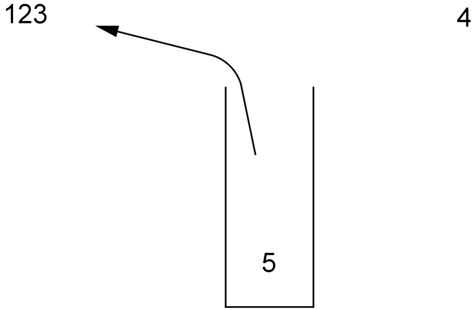
STACK SORTING

Stack sorting is a method of sorting a permutation where we are only allowed to use two operations: Put the next element from the input onto the stack, and output the top element of the stack.



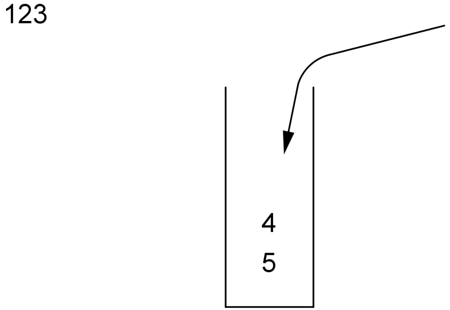
STACK SORTING

Stack sorting is a method of sorting a permutation where we are only allowed to use two operations: Put the next element from the input onto the stack, and output the top element of the stack.



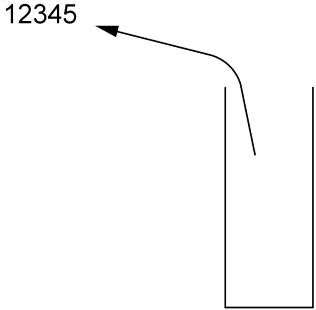
Stack Sorting

Stack sorting is a method of sorting a permutation where we are only allowed to use two operations: Put the next element from the input onto the stack, and output the top element of the stack.



Stack Sorting

Stack sorting is a method of sorting a permutation where we are only allowed to use two operations: Put the next element from the input onto the stack, and output the top element of the stack.



Stack Sorting

Donald Knuth introduced stack sorting in The Art of Computer Programming.

Stack Sorting

Donald Knuth introduced stack sorting in The Art of Computer Programming.

Knuth also showed that these permutations are counted by the Catalan numbers

$$\frac{1}{n+1}\binom{2n}{n}$$

Stack Sorting

Donald Knuth introduced stack sorting in The Art of Computer Programming.
Knuth also showed that these permutations are counted by the Catalan numbers

$$\frac{1}{n+1}\binom{2n}{n}$$

This is because each sequence of moves corresponds to sorting a different permutation.

Stack Sorting

Donald Knuth introduced stack sorting in The Art of Computer Programming.
Knuth also showed that these permutations are counted by the Catalan numbers

$$\frac{1}{n+1}\binom{2n}{n}$$

This is because each sequence of moves corresponds to sorting a different permutation.
He also proved that the set of stack-sortable permutations is Av(312).

Other Sorting Machines

Other Sorting Machines

Knuth also considered a few other sorting machines, including:

Other Sorting Machines

Knuth also considered a few other sorting machines, including:

- Two stacks in parallel

Other Sorting Machines

Knuth also considered a few other sorting machines, including:

- Two stacks in parallel
- A double ended queue (deque)

Other Sorting Machines

Knuth also considered a few other sorting machines, including:

- Two stacks in parallel
- A double ended queue (deque)
- Two stacks in series

OTHER SORTING MACHINES

Knuth also considered a few other sorting machines, including:

- Two stacks in parallel
- A double ended queue (deque)
- Two stacks in series

He posed the question of how many permutations of each length can be sorted by each of these machines.

Other Sorting Machines

Knuth also considered a few other sorting machines, including:

- Two stacks in parallel
- A double ended queue (deque)
- Two stacks in series

He posed the question of how many permutations of each length can be sorted by each of these machines.
Until 2015 none of these two questions had been solved.

Two stacks in parallel

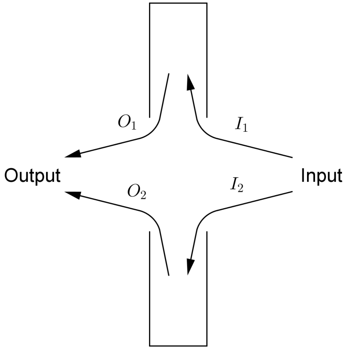
TWO STACKS IN PARALLEL

With two stacks in parallel, it is permitted to either move an element from the input onto either of the two stacks, or output the top element from either of the stacks.

Two stacks in parallel

With two stacks in parallel, it is permitted to either move an element from the input onto either of the two stacks, or output the top element from either of the stacks.

We call these operations I_1, I_2, O_1, O_2 as shown:



TWO STACKS IN PARALLEL

Michael Albert and Mireille Bousquet-Melou solved this problem in 2015, showing that the associated generating function is the solution of a series of functional equations.

Two stacks in parallel

Michael Albert and Mireille Bousquet-Melou solved this problem in 2015, showing that the associated generating function is the solution of a series of functional equations.
I will give a summary of their solution.

SOLUTION FOR TWO STACKS IN PARALLEL

- Call a sequence of the moves which corresponds to sorting a permutation an “operation sequence".

Solution for two stacks in parallel

- Call a sequence of the moves which corresponds to sorting a permutation an “operation sequence".
- By associating the operations I_1, I_2, O_1, O_2 with the steps N, E, S, W, each operation sequence corresponds to a unique quarter plane loop.

Solution for two stacks in parallel

- Call a sequence of the moves which corresponds to sorting a permutation an "operation sequence".
- By associating the operations I_1, I_2, O_1, O_2 with the steps N, E, S, W, each operation sequence corresponds to a unique quarter plane loop.
- Unfortunately, many different operation sequences correspond to the same permutation.

Solution for two stacks in parallel

- Call a sequence of the moves which corresponds to sorting a permutation an “operation sequence".
- By associating the operations I_1, I_2, O_1, O_2 with the steps N, E, S, W, each operation sequence corresponds to a unique quarter plane loop.
- Unfortunately, many different operation sequences correspond to the same permutation.
- Call two operation sequences equivalent if they sort the same permutation.

SOLUTION FOR TWO STACKS IN PARALLEL

- Call a sequence of the moves which corresponds to sorting a permutation an “operation sequence".
- By associating the operations I_1, I_2, O_1, O_2 with the steps N, E, S, W, each operation sequence corresponds to a unique quarter plane loop.
- Unfortunately, many different operation sequences correspond to the same permutation.
- Call two operation sequences equivalent if they sort the same permutation.
- We want to count the number of equivalence classes of operation sequences.

Solution for two stacks in parallel

Albert and Bousquet-Melou found a canonical class of operation sequences, which contains exactly one operation sequence in each equivalence class.

SOLUTION FOR TWO STACKS IN PARALLEL

Albert and Bousquet-Melou found a canonical class of operation sequences, which contains exactly one operation sequence in each equivalence class.
This canonical class contains the operation sequences which satisfy the following two conditions:

SOLUTION FOR TWO STACKS IN PARALLEL

Albert and Bousquet-Melou found a canonical class of operation sequences, which contains exactly one operation sequence in each equivalence class.
This canonical class contains the operation sequences which satisfy the following two conditions:

- The subwords I_1O_2 and I_2O_1 are forbidden.

Solution for two stacks in parallel

Albert and Bousquet-Melou found a canonical class of operation sequences, which contains exactly one operation sequence in each equivalence class.
This canonical class contains the operation sequences which satisfy the following two conditions:

- The subwords I_1O_2 and I_2O_1 are forbidden.
- Any subword of the operation sequence which is also an operation sequence begins with I_1.

SOLUTION FOR TWO STACKS IN PARALLEL

Albert and Bousquet-Melou found a canonical class of operation sequences, which contains exactly one operation sequence in each equivalence class.
This canonical class contains the operation sequences which satisfy the following two conditions:

- The subwords I_1O_2 and I_2O_1 are forbidden.
- Any subword of the operation sequence which is also an operation sequence begins with I_1.

This reduces the problem to counting a certain class of quarter plane loops: Those with no *NW* or *ES* corners, in which every sub-(quarter plane loop) begins with *N*.

Solution for two stacks in parallel

SOLUTION FOR TWO STACKS IN PARALLEL

They related $P(t)$, the counting function of tsip-sortable permutation, to the generating function $Q(a,u)$ for weighted quarter plane loops, where a counts the number of NW or ES corners, and u counts the halflength.

SOLUTION FOR TWO STACKS IN PARALLEL

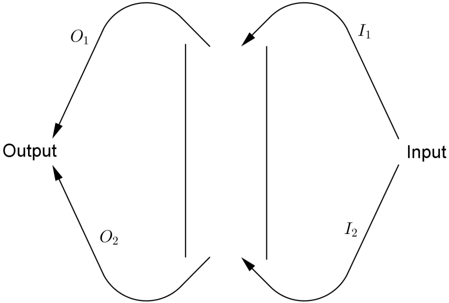
They related $P(t)$, the counting function of tsip-sortable permutation, to the generating function $Q(a, u)$ for weighted quarter plane loops, where a counts the number of NW or ES corners, and u counts the halflength.
In particular, they showed that $P(t)$ is characterised by the equation

$$Q\left(\frac{1}{P} - 1, \frac{tP^2}{(2P-1)^2}\right) = 2P - 1$$

Sorting With a Deque

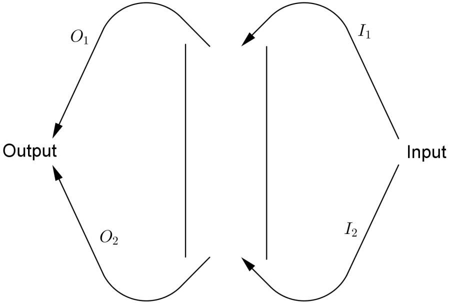
Sorting With a Deque

With a deque, There are also four moves I_1, I_2, O_1, O_2, which correspond to input to the top and bottom of the deque and output from the top and bottom of the deque.



Sorting With a Deque

With a deque, There are also four moves I_1, I_2, O_1, O_2, which correspond to input to the top and bottom of the deque and output from the top and bottom of the deque.



Notice that if we put a wall in the middle of the deque, this becomes equivalent to two stacks in parallel.

SORTING WITH A DEQUE

Sorting With a Deque

Guttmann and I were able to adapt the solution for two stacks in parallel to solve the enumeration problem for deque-sortable permutations:

Sorting With a Deque

Guttmann and I were able to adapt the solution for two stacks in parallel to solve the enumeration problem for deque-sortable permutations:

- We associate the operations I_1, I_2, O_1, O_2 with the steps N, E, S, W, respectively.

Sorting With a Deque

Guttmann and I were able to adapt the solution for two stacks in parallel to solve the enumeration problem for deque-sortable permutations:

- We associate the operations I_1, I_2, O_1, O_2 with the steps N, E, S, W, respectively.
- The corresponding path of an operation sequence is now restricted to the diagonal half plane $\{(x, y) | x + y \geq 0\}$, and must end on the line $x + y = 0$.

Sorting With a Deque

We defined the canonical operation sequence of a deque-sortable permutation to be the first in the lexicographic order with $O_1 < O_2 < I_1 < I_2$.

Sorting With a Deque

We defined the canonical operation sequence of a deque-sortable permutation to be the first in the lexicographic order with
$O_1 < O_2 < I_1 < I_2$.
We then proved that canonical operation sequences are exactly the operation sequences with the following properties:

We defined the canonical operation sequence of a deque-sortable permutation to be the first in the lexicographic order with $O_1 < O_2 < I_1 < I_2$.
We then proved that canonical operation sequences are exactly the operation sequences with the following properties:

- The subwords $I_1 O_2$ and $I_2 O_1$ are forbidden.

Sorting With a Deque

We defined the canonical operation sequence of a deque-sortable permutation to be the first in the lexicographic order with $O_1 < O_2 < I_1 < I_2$.

We then proved that canonical operation sequences are exactly the operation sequences with the following properties:

- The subwords I_1O_2 and I_2O_1 are forbidden.
- Any subword of the operation sequence which is an operation sequence for two stacks in parallel begins with I_1.

SORTING WITH A DEQUE

We defined the canonical operation sequence of a deque-sortable permutation to be the first in the lexicographic order with $O_1 < O_2 < I_1 < I_2$.
We then proved that canonical operation sequences are exactly the operation sequences with the following properties:

- The subwords I_1O_2 and I_2O_1 are forbidden.
- Any subword of the operation sequence which is an operation sequence for two stacks in parallel begins with I_1.
- When the deque contains at most 1 element only the moves I_1 and O_1 are allowed, not I_2 or O_2.

SORTING WITH A DEQUE

Rewriting the last slide, $D(t)$ count walks w in the diagonal half plane such that:

SORTING WITH A DEQUE

Rewriting the last slide, $D(t)$ count walks w in the diagonal half plane such that:

- w starts at $(0,0)$ and ends on the line $x+y=0$.

SORTING WITH A DEQUE

Rewriting the last slide, $D(t)$ count walks w in the diagonal half plane such that:

- w starts at $(0,0)$ and ends on the line $x+y=0$.
- w contains no NW or ES corners.

Sorting With a Deque

Rewriting the last slide, $D(t)$ count walks w in the diagonal half plane such that:

- w starts at $(0,0)$ and ends on the line $x+y=0$.
- w contains no NW or ES corners.
- Any sub path of w which forms a quarter plane loop begins with N.

Rewriting the last slide, $D(t)$ count walks w in the diagonal half plane such that:

- w starts at $(0,0)$ and ends on the line $x+y=0$.
- w contains no NW or ES corners.
- Any sub path of w which forms a quarter plane loop begins with N.
- Any step from a point (x,y) with $x+y \leq 1$ must be N or S.

SORTING WITH A DEQUE

- We characterised $D(t)$ by relating it to a series of functional equations including P and Q...

SORTING WITH A DEQUE

- We characterised $D(t)$ by relating it to a series of functional equations including P and Q...
- Then we cancelled some functions from the equations...

SORTING WITH A DEQUE

- We characterised $D(t)$ by relating it to a series of functional equations including P and Q...
- Then we cancelled some functions from the equations...
- Finally, we deduced the remarkable result that the generating functions D for deques and P for two stacks in Parallel are related by the equation

$$D(t) = \frac{t}{2} + 1 + tP - t^2P - \frac{t}{2}\sqrt{1 - 4P + 4P^2 - 8tP^2 + 4t^2P^2 - 4tP}$$

QUARTER PLANE LOOPS

QUARTER PLANE LOOPS

To complete the solution of the two problems, we need to show that the problem of enumerating weighted quarter plane loops is “solved":

QUARTER PLANE LOOPS

To complete the solution of the two problems, we need to show that the problem of enumerating weighted quarter plane loops is "solved":

- First, let $W(a, u, x, y)$ be the generating function for weighted quarter plane *walks*, given by

  $$\sum w_{m,n,\alpha,\beta} a^m u^n x^\alpha y^\beta,$$

  where $w_{m,n,\alpha,\beta}$ is the number of quarter plane walks from $(0, 0)$ to (α, β) of length n which contain m weighted corners.

Quarter plane loops

To complete the solution of the two problems, we need to show that the problem of enumerating weighted quarter plane loops is “solved":

- First, let $W(a, u, x, y)$ be the generating function for weighted quarter plane *walks*, given by

$$\sum w_{m,n,\alpha,\beta} a^m u^n x^\alpha y^\beta,$$

  where $w_{m,n,\alpha,\beta}$ is the number of quarter plane walks from $(0, 0)$ to (α, β) of length n which contain m weighted corners.
- Then W is characterised by the equation

$$\begin{aligned} W(x,y) = u(x+y)W(x,y) &+ \frac{u}{y}(1 - ux + uax)(W(x,y) - W(x,0)) \\ &+ \frac{u}{x}(1 - uy + uay)(W(x,y) - W(0,y)). \end{aligned}$$

QUARTER PLANE LOOPS

To complete the solution of the two problems, we need to show that the problem of enumerating weighted quarter plane loops is “solved":

- First, let $W(a, u, x, y)$ be the generating function for weighted quarter plane *walks*, given by

$$\sum w_{m,n,\alpha,\beta} a^m u^n x^\alpha y^\beta,$$

  where $w_{m,n,\alpha,\beta}$ is the number of quarter plane walks from $(0, 0)$ to (α, β) of length n which contain m weighted corners.
- Then W is characterised by the equation

$$\begin{aligned} W(x,y) = u(x+y)W(x,y) &+ \frac{u}{y}(1 - ux + uax)(W(x,y) - W(x,0)) \\ &+ \frac{u}{x}(1 - uy + uay)(W(x,y) - W(0,y)). \end{aligned}$$

- Finally $Q(a, u) = W(a, \sqrt{u}, 0, 0)$.

Analysis for two stacks in parallel

Analysis for two stacks in parallel

Using the equation

$$Q\left(\frac{1}{P}-1,\frac{tP^2}{(2P-1)^2}\right)=2P-1,$$

we wrote a program which calculated the first 1337 coefficients of the series $P(t)$.

Analysis for two stacks in parallel

Using the equation

$$Q\left(\frac{1}{P}-1,\frac{tP^2}{(2P-1)^2}\right)=2P-1,$$

we wrote a program which calculated the first 1337 coefficients of the series $P(t)$.
We tried to Jay it...

Analysis for two stacks in parallel

Using the equation

$$Q\left(\frac{1}{P}-1,\frac{tP^2}{(2P-1)^2}\right)=2P-1,$$

we wrote a program which calculated the first 1337 coefficients of the series $P(t)$.
We tried to Jay it... But it was not Jayable.

Analysis for two stacks in parallel

Using the equation

$$Q\left(\frac{1}{P}-1,\frac{tP^2}{(2P-1)^2}\right)=2P-1,$$

we wrote a program which calculated the first 1337 coefficients of the series $P(t)$.

We tried to Jay it... But it was not Jayable. (Jay Pantone did a search using these terms for a differentially algebraic solution but did not find one.)

Analysis for two stacks in parallel

Using the equation

$$Q\left(\frac{1}{P}-1,\frac{tP^2}{(2P-1)^2}\right)=2P-1,$$

we wrote a program which calculated the first 1337 coefficients of the series $P(t)$.
We tried to Jay it... But it was not Jayable. (Jay Pantone did a search using these terms for a differentially algebraic solution but did not find one.)
Our analysis of the coefficients p_n of P suggests that they behave like

$$p_n \sim const \cdot \mu^n n^\gamma,$$

where $\mu \approx 8.281402207$ and $\gamma \approx -2.473$.

ANALYSIS OF $D(t)$

ANALYSIS OF $D(t)$

$$D(t) = \frac{t}{2} + 1 + tP - t^2P - \frac{t}{2}\sqrt{1 - 4P + 4P^2 - 8tP^2 + 4t^2P^2 - 4tP}$$

ANALYSIS OF $D(t)$

$$D(t) = \frac{t}{2} + 1 + tP - t^2P - \frac{t}{2}\sqrt{1 - 4P + 4P^2 - 8tP^2 + 4t^2P^2 - 4tP}$$

We found the first 1337 coefficients of the generating function $D(t)$ using those of $P(t)$.

ANALYSIS OF $D(t)$

$$D(t) = \frac{t}{2} + 1 + tP - t^2P - \frac{t}{2}\sqrt{1 - 4P + 4P^2 - 8tP^2 + 4t^2P^2 - 4tP}$$

We found the first 1337 coefficients of the generating function $D(t)$ using those of $P(t)$.
Our analysis of these terms suggests that the coefficients behave like

$$d_n \sim const \cdot \mu^n n^{-3/2},$$

where $\mu \approx 8.281402207$ is the same as the growth constant for the coefficients of $P(t)$.

Analysis of $D(t)$

$$D(t) = \frac{t}{2} + 1 + tP - t^2P - \frac{t}{2}\sqrt{1 - 4P + 4P^2 - 8tP^2 + 4t^2P^2 - 4tP}$$

We found the first 1337 coefficients of the generating function $D(t)$ using those of $P(t)$.
Our analysis of these terms suggests that the coefficients behave like

$$d_n \sim const \cdot \mu^n n^{-3/2},$$

where $\mu \approx 8.281402207$ is the same as the growth constant for the coefficients of $P(t)$.
We still can't prove that the two squences $\{p_n\}$ and $\{d_n\}$ share the same exponential growth rate μ.

MORE ANALYSIS

More analysis

Assuming a few conjectures about the generating function $Q(a,u)$ for weighted quarterplane loops, Albert and Bousquet-Melou proved the following about $P(t)$:

More analysis

Assuming a few conjectures about the generating function $Q(a, u)$ for weighted quarterplane loops, Albert and Bousquet-Melou proved the following about $P(t)$:

$$\sqrt{2P(t_c)} = 1 + \sqrt{2t_cP(t_c)},$$

where t_c is the radius of convergence of P.

More analysis

Assuming a few conjectures about the generating function $Q(a,u)$ for weighted quarterplane loops, Albert and Bousquet-Melou proved the following about $P(t)$:

$$\sqrt{2P(t_c)} = 1 + \sqrt{2t_cP(t_c)},$$

where t_c is the radius of convergence of P.
Then, using our equation

$$D(t) = \frac{t}{2} + 1 + tP - t^2P - \frac{t}{2}\sqrt{1 - 4P + 4P^2 - 8tP^2 + 4t^2P^2 - 4tP}$$

it follows pretty easily that the radius of convergence of $D(t)$ is also t_c...

MORE ANALYSIS

Assuming a few conjectures about the generating function $Q(a, u)$ for weighted quarterplane loops, Albert and Bousquet-Melou proved the following about $P(t)$:

$$\sqrt{2P(t_c)} = 1 + \sqrt{2t_cP(t_c)},$$

where t_c is the radius of convergence of P.
Then, using our equation

$$D(t) = \frac{t}{2} + 1 + tP - t^2P - \frac{t}{2}\sqrt{1 - 4P + 4P^2 - 8tP^2 + 4t^2P^2 - 4tP}$$

it follows pretty easily that the radius of convergence of $D(t)$ is also t_c...**But we still need to prove the conjectures about $Q(a, u)$!!!!**

Quarter plane loops conjecture 1

Quarter plane loops conjecture 1

Conjecture

the radius of convergence $\rho_Q(a)$ of $Q(a,\cdot)$ is given by

$$\rho_Q(a) = \begin{cases} \dfrac{1}{(2+\sqrt{2+2a})^2}, & \text{if } a \geq -1/2, \\ \dfrac{-a}{2(a-1)^2}, & \text{if } a \in [-1,-1/2]. \end{cases}$$

Quarter plane loops conjecture 1

Conjecture

the radius of convergence $\rho_Q(a)$ of $Q(a,\cdot)$ is given by

$$\rho_Q(a) = \begin{cases} \dfrac{1}{(2+\sqrt{2+2a})^2}, & \textit{if } a \geq -1/2, \\ \dfrac{-a}{2(a-1)^2}, & \textit{if } a \in [-1,-1/2]. \end{cases}$$

For $a \geq -1/2$, this is the same as the radius of convergence for weighted plane paths (not just **quarter**-plane **loops**).

Quarter plane loops conjecture 1

Conjecture

the radius of convergence $\rho_Q(a)$ of $Q(a,\cdot)$ is given by

$$\rho_Q(a) = \begin{cases} \dfrac{1}{(2+\sqrt{2+2a})^2}, & \text{if } a \geq -1/2, \\ \dfrac{-a}{2(a-1)^2}, & \text{if } a \in [-1,-1/2]. \end{cases}$$

For $a \geq -1/2$, this is the same as the radius of convergence for weighted plane paths (not just **quarter**-plane **loops**).

The really important bit of the conjecture is at $a \approx -0.148$.

QUARTER PLANE LOOPS CONJECTURES 2 AND 3

Quarter plane loops conjectures 2 and 3

Conjecture

The series $Q(a,u)$ is $(a+1)$-positive. That is, Q takes the form

$$Q(a,u) = \sum_{n \geq 0} u^n P_n(a+1),$$

where each polynomial P_n has positive coefficients.

Quarter plane loops conjectures 2 and 3

Conjecture

The series $Q(a, u)$ is $(a+1)$-positive. That is, Q takes the form

$$Q(a,u) = \sum_{n \geq 0} u^n P_n(a+1),$$

where each polynomial P_n has positive coefficients.

Conjecture

The series $Q_u(a,u) = \frac{\partial Q}{\partial u}$ is convergent at $u = \rho_Q(a)$ for $a \geq -1/3$.

Open problems

OPEN PROBLEMS

- Prove that Permutations sortable by a deque and permutations sortable by two stacks in parallel have the same growth constant!

Open problems

- Prove that Permutations sortable by a deque and permutations sortable by two stacks in parallel have the same growth constant!
- Just prove any of the three conjectures about weighted quarter plane loops.

Open problems

- Prove that Permutations sortable by a deque and permutations sortable by two stacks in parallel have the same growth constant!
- Just prove any of the three conjectures about weighted quarter plane loops.
- Find a more direct proof of the identity

$$D(t) = \frac{t}{2}+1+tP-t^2P-\frac{t}{2}\sqrt{1-4P+4P^2-8tP^2+4t^2P^2-4tP}$$

OPEN PROBLEMS

OPEN PROBLEMS

The following machines are still unsolved:

OPEN PROBLEMS

The following machines are still unsolved:

- Three stacks in parallel

OPEN PROBLEMS

The following machines are still unsolved:

- Three stacks in parallel
- Four stacks in parallel

OPEN PROBLEMS

The following machines are still unsolved:

- Three stacks in parallel
- Four stacks in parallel
- etc

Open problems

The following machines are still unsolved:

- Three stacks in parallel
- Four stacks in parallel
- etc
- Two stacks in series

Open problems

The following machines are still unsolved:

- Three stacks in parallel
- Four stacks in parallel
- etc
- Two stacks in series
- Two input-restricted deques in parallel

OPEN PROBLEMS

The following machines are still unsolved:

- Three stacks in parallel
- Four stacks in parallel
- etc
- Two stacks in series
- Two input-restricted deques in parallel
- One input-restricted deque and one stack in parallel

Open problems

The following machines are still unsolved:

- Three stacks in parallel
- Four stacks in parallel
- etc
- Two stacks in series
- Two input-restricted deques in parallel
- One input-restricted deque and one stack in parallel
- Any finite system of machines in parallel/series which involves any of the above

THANK YOU

Thank You!