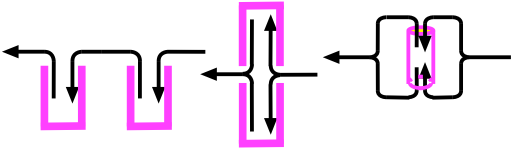
Computing Permutations with Stacks and Deques

Michael Albert[1] Mike Atkinson[1] Steve Linton[2]

[1]Department of Computer Science, University of Otago

[2]School of Computer Science, University of St Andrews

7th Australia – New Zealand Mathematics Convention



Outline of talk

Data Structures

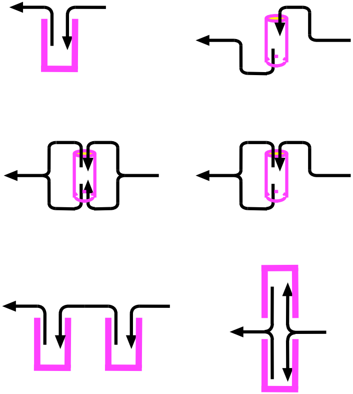
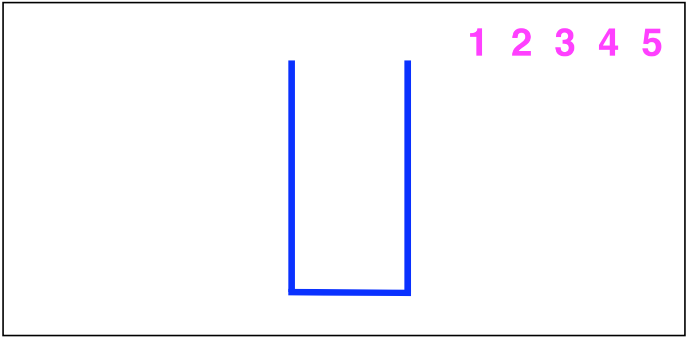
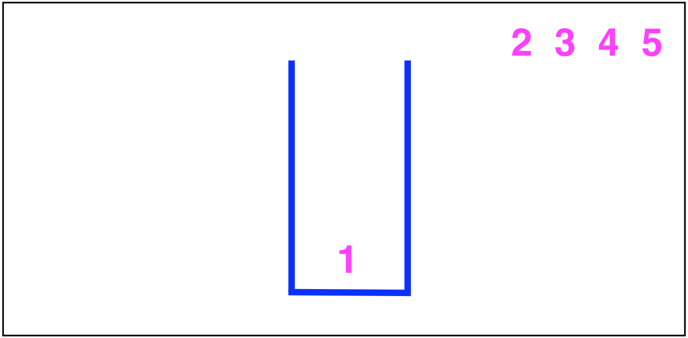


Figure: What permutations can a data structure generate (or sort)?

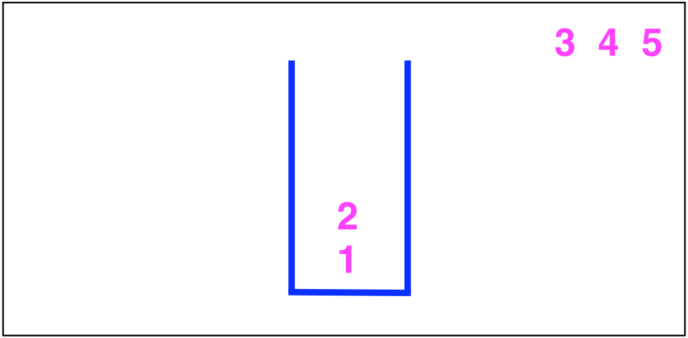
Generating a permutation



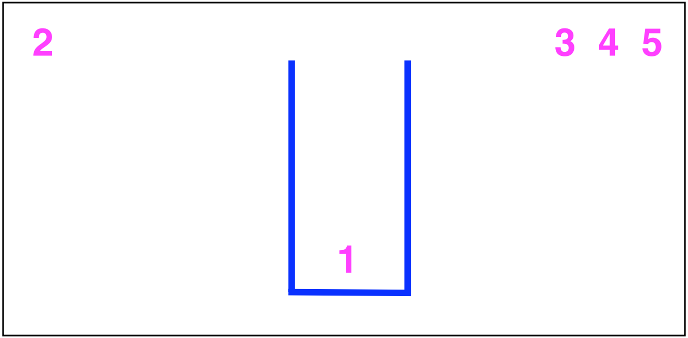
Generating a permutation



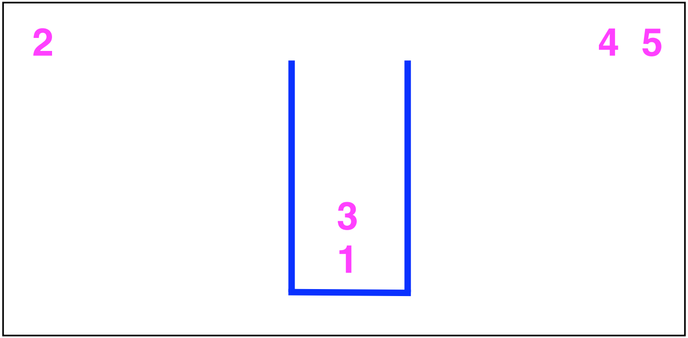
Generating a permutation



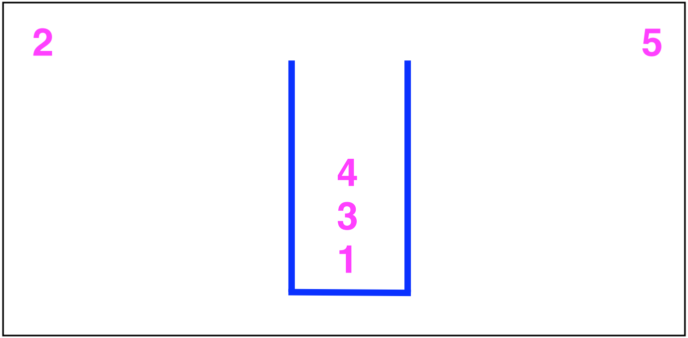
Generating a permutation



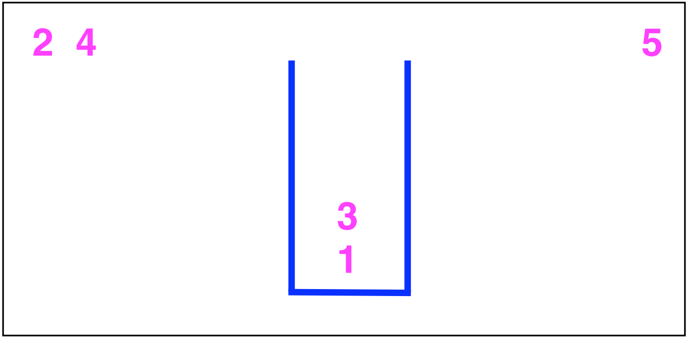
Generating a permutation



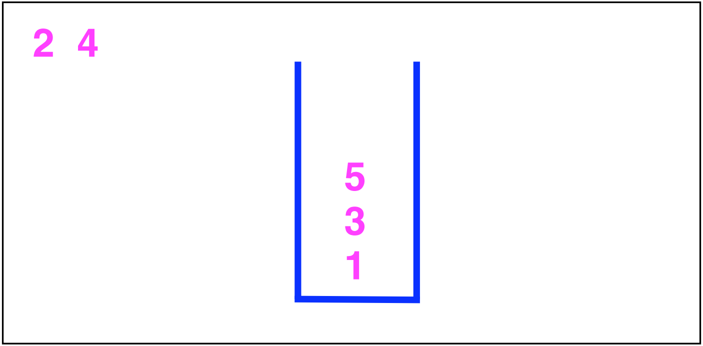
Generating a permutation



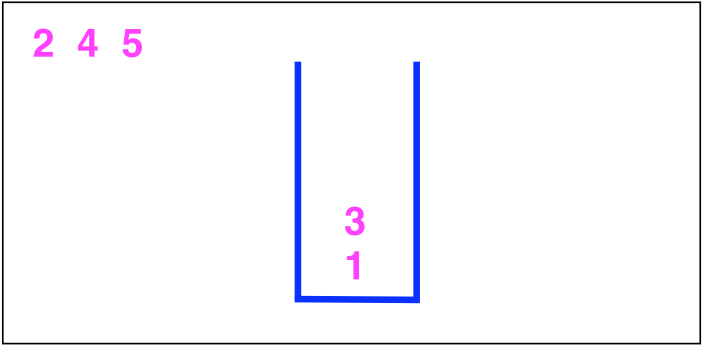
Generating a permutation



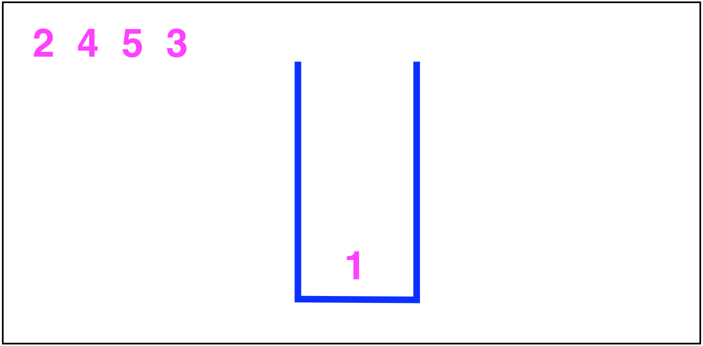
Generating a permutation



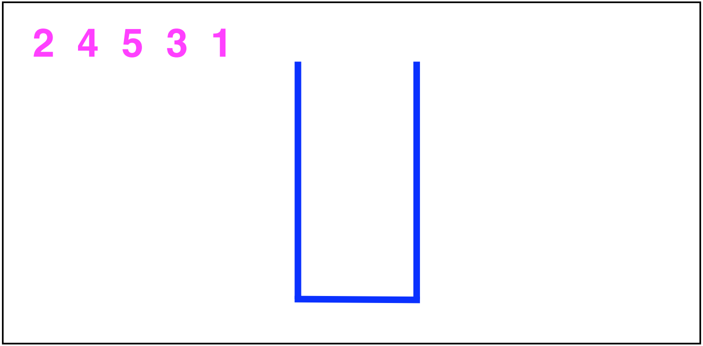
Generating a permutation



Generating a permutation



Generating a permutation



General question

Question

How many permutations can some given data structure generate?

Donald Knuth

Knuth: results

- There is one “queue permutation” of every length

Knuth: results

- There is one "queue permutation" of every length
- There are

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

"stack permutations" of length n

Knuth: results

- There is one "queue permutation" of every length
- There are

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

  "stack permutations" of length n
- There are r_n "restricted input deque permutations" of length n where

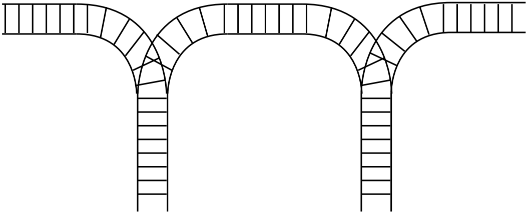
$$\sum_{n=1}^{\infty} r_n x^n = \frac{1 - x - \sqrt{1 - 6x + x^2}}{2}$$

Knuth: questions

- Exercise 2.2.1.13: "*[M48] How many permutations of n elements are obtainable with the use of a general deque?*"

Knuth: questions

- Exercise 2.2.1.13: "*[M48] How many permutations of n elements are obtainable with the use of a general deque?*"
- What about stacks in series? In parallel?



Data Structures

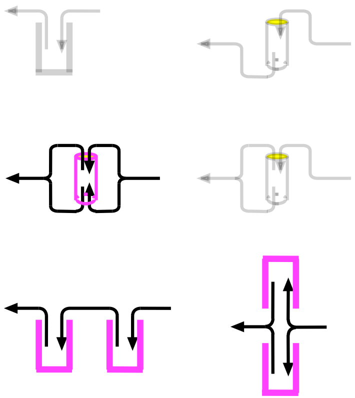


Figure: Data Structures with unknown enumerations

Growth rates

- We can't always do exact counting
- Approximate the exact count of permutations of length n by γ^n; γ is called the growth rate
- Eg. For stacks

$$\frac{\binom{2n}{n}}{n+1} \sim \frac{4^n}{n^{3/2}}$$

so growth rate 4.

- The growth rate of a sequence (c_n) is formally defined as

$$\gamma = \limsup_{n\to\infty} \sqrt[n]{c_n}$$

The research question

Question

What is the growth rate for deques, two stacks in parallel, two stacks in series?

It is known that, in all three cases, the growth rate is between 4 and 16.

Regular sets: $p(qq^*p)^*q^*$

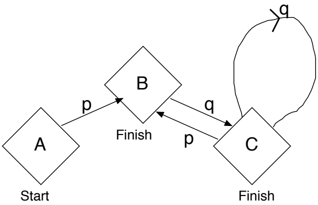


Figure: Recognises strings of p's and q's beginning with p with no pp

Regular sets: $p(qq^*p)^*q^*$

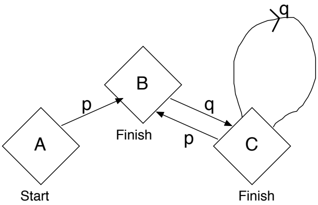


Figure: Recognises strings of p's and q's beginning with p with no pp

$$A = 1; B = Ax + Cx; C = Cx + Bx$$

Regular sets: $p(qq^*p)^*q^*$

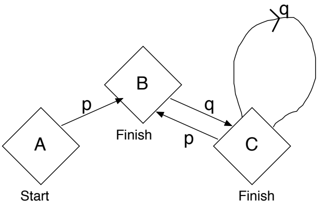


Figure: Recognises strings of p's and q's beginning with p with no pp

$$A = 1; B = Ax + Cx; C = Cx + Bx$$

$$B + C = \frac{x}{1 - x - x^2} = x + x^2 + 2x^3 + 3x^4 + 5x^5 + 8x^6 + 13x^7 \cdots$$

The general algebraic method

- Start from a FSM (or regular set of strings)
- Mechanically produce the "generating function" $A(x)$
- The form of $A(x)$ is always a quotient of two polynomials $p(x)$ and $q(x)$

$$A(x) = \frac{p(x)}{q(x)}$$

- Either
  - Expand $A(x)$ as a power series $a_0 + a_1 x + a_2 x^2 + \cdots$ and find a_n, the number of strings of length n, or
  - Find the growth rate of a_n by solving $q(x) = 0$

So counting is easy if we begin from a regular set.

Upper bounds on growth rates – Stacks in series

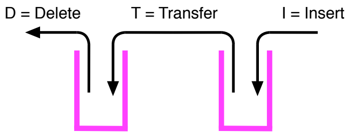


Figure: Two stacks in series

- *IIITITDTDDTD* produces 4231 from input 1234

Upper bounds on growth rates – Stacks in parallel

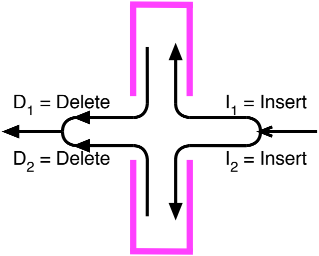


Figure: Two stacks in parallel

- $I_1 I_1 I_2 D_1 I_2 I_1 D_2 I_1 D_2 D_2 D_1 D_1$ produces 24351 from input 12345

Upper bounds on growth rates – Deques

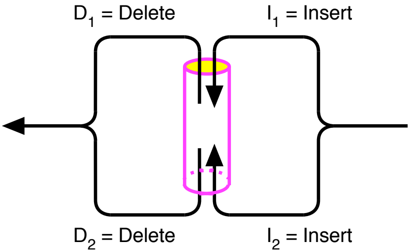


Figure: Deque

- $I_1 I_2 I_1 D_2 I_2 I_2 I_1 D_2 D_1 D_2 D_2 D_2 D_1$ produces 256413

Permutations as strings

- Represent permutations by strings over a 3 or 4 letter alphabet and count strings. This is an overcount since
  1. Not every string represents a permutation, and
  2. Many strings represent the same permutation
- The first of these doesn't seem to matter much for growth rates. The second is much more serious.

Rewriting rules

Definition

If L, R are strings then $L \to R$ if any permutation which can be generated by a string ULV is also generated by URV.

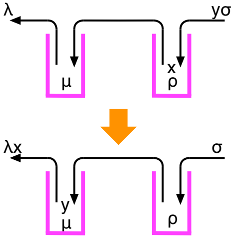


Figure: $TDIT \to ITTD$

Getting upper bounds

- Systematically collect as many rewriting rules as you can
- Count strings of length n that have no LHS as a substring
- This is a count of strings in a regular set!

Results – Deque

Length	Number of Rules	Growth Upper Bound
8	51	8.4925
9	85	8.459
10	175	8.428
11	321	8.410
12	756	8.392
13	1480	8.380
14	3806	8.368
15	7734	8.361
16	21029	8.352

Results – Parallel Stacks

Length	Number of Rules	Growth Upper Bound
8	33	8.4606
9	43	8.4474
10	109	8.4087
11	143	8.4031
12	466	8.379
13	615	8.376
14	2366	8.3597
15	3131	8.3578
16	13263	8.3461

Results – Two Stacks in Series

Length	Number of Rules	Growth Upper Bound
8	23	14.201
9	35	14.048
10	71	13.826
11	106	13.747
12	215	13.623
13	368	13.552
14	737	13.477
15	1270	13.433
16	2825	13.374

Lower bounds – via bounded capacities

- Consider k-bounded versions of the three structures where the system is constrained to contain at most k elements at a time.

Lower bounds – via bounded capacities

- Consider k-bounded versions of the three structures where the system is constrained to contain at most k elements at a time.
- The system can now be thought of as FSA with states that correspond to the disposition of elements residing in the stacks/deque.

Lower bounds – via bounded capacities

- Consider k-bounded versions of the three structures where the system is constrained to contain at most k elements at a time.
- The system can now be thought of as FSA with states that correspond to the disposition of elements residing in the stacks/deque.
- It outputs rank-encoded permutations: e.g. 4163752 is encoded as 4142321 – and the ranks will be at most k

Bounded deque FSA

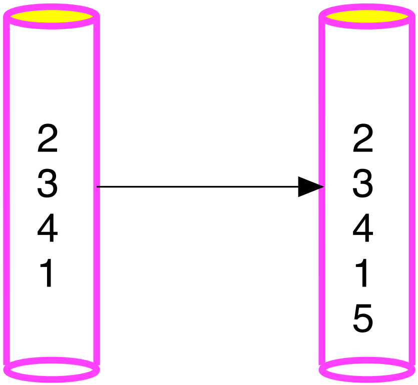


Figure: The deque FSA when a symbol is added to the bottom

Bounded deque FSA

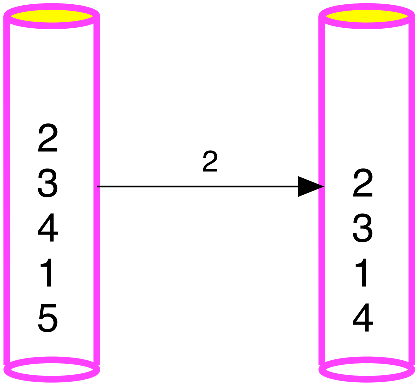


Figure: The deque FSA when a symbol is removed from the top

Getting lower bounds

- Compute the non-deterministic FSA for a k-bounded system
- Compute the corresponding deterministic automaton
- Compute the growth rate of the k-bounded system which will be a lower bound for the growth rate of the unrestricted system
- Many tricks to contain the state explosion

Results

	k	Growth Lower Bound
Serial stacks	9	8.156
Parallel stacks	18	7.535
Deques	21	7.890

Bottom line for growth rate γ

- Two stacks in series: $8.156 \leq \gamma \leq 13.374$
- Two stacks in parallel: $7.535 \leq \gamma \leq 8.3461$
- Deque: $7.890 \leq \gamma \leq 8.352$

Open questions

- What are the true growth rates?
- Do deques and two parallel stacks have the same growth rate?
- Why is two stacks in series more difficult?
- For deques and two parallel stacks we have efficient recognition algorithms; is the recognition problem for two stacks in series NP-complete?
- Can we get the *exact* enumerations for two parallel stacks? For deques?