Gray Codes and Universal Cycles

minisymposium

Joe Sawada	An overview of combinatorial generation
Aaron Williams	Iterative Gray codes
Ryuhei Uehara	On generation of graphs with geometric representations
Xi Sisi Shen	A *Hot Potato* transposition Gray code for permutations
~~*Victoria Horan*~~	~~*Universal cycles for weak orders*~~

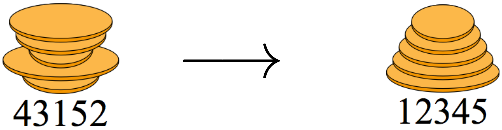
Whaddya at? St. John's June 11, 2013
CANADAM

The "Harried" Waiter Problem

A waiter in the local **IHOP** (International House of Pancakes) is running around with a stack of n different sized pancakes. To reduce the chance of a disaster, he wants to **sort** the pancakes by size. But with only one free hand, and a spatula, the only operation he can perform is flips of various sizes.

What is the maximum number of flips he will require?

– *Harry Dweighter* (John Goodman, 1975)

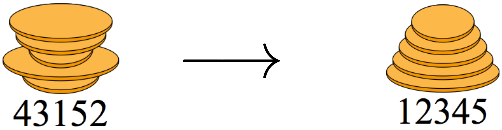


The "Harried" Waiter Problem

A waiter in the local **IHOP** (International House of Pancakes) is running around with a stack of n different sized pancakes. To reduce the chance of a disaster, he wants to **sort** the pancakes by size. But with only one free hand, and a spatula, the only operation he can perform is flips of various sizes.

What is the maximum number of flips he will require?

– *Harry Dweighter* (John Goodman, 1975)



Corresponding Problem – NP-hard Bulteau, Fertin, Rusu (2012)

What is the maximum number of **prefix-reversals** (flips) required to sort a permutation into ascending order?

The "Intrepid" Waitress Problems

An **intrepid waitress** in the local IHOP is delivering a stack of n different sized pancakes. Instead of simply sorting the pancakes, she wants to iterate through all possible stack orderings.

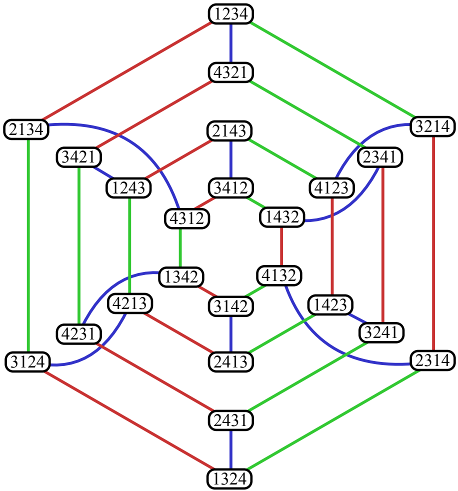
Can the waitress iterate through all possible stacks of pancakes?

Example: $n = 5$

Corresponding Problem – Zaks (1984)

Can you construct a **listing of all** n! **stacks** of n pancakes (permutations) such that successive permutations differ by a single flip?

Pancake Network $n = 4$



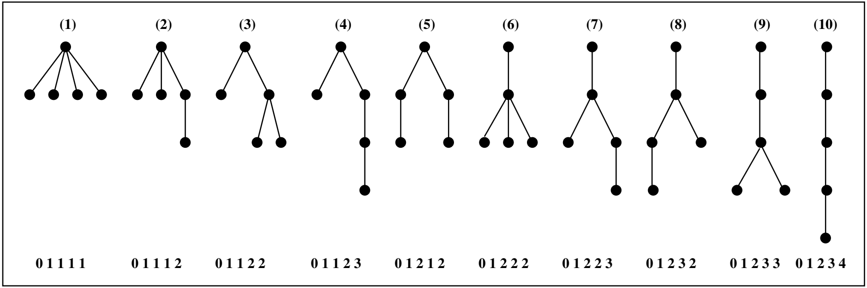
Corresponding Problem

Is there a **Hamilton path** (cycle) in the Pancake Network?

Combinatorial Generation

Primary Goal

Given a combinatorial object (permutations, trees, necklaces, graphs), find an **efficient** algorithm to exhaustively list each instance exactly once



Important considerations

- Representation
- Ordering: lexicographic, Gray code

Related issues

- Enumeration
- Random generation
- Ranking, unranking

Such algorithms are often very short but hard to locate and usually are surprisingly subtle

– Steven Skiena, The Stony Brook Algorithm Repository

Gray Codes

The Binary Reflected Gray Code was patented in 1953 by Frank Gray.

$$\mathrm{BRGC}(n) \;=\; \mathbf{0} \cdot \mathrm{BRGC}(n-1),\; \mathbf{1} \cdot \mathrm{BRGC}(n-1)^R$$

BRGC(3) =

0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

Gray Code Order

Successive objects in an exhaustive listing differ by a single operation (or constant amount of work)

Volume Control Application

Boom Box Volume

Lexicographic Order

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Gray Code Order

0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

Volume Control Application

Boom Box Volume

Lexicographic Order

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

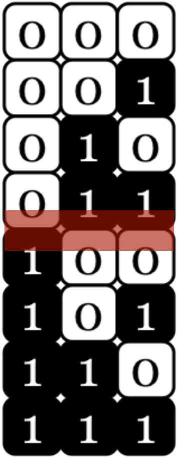
Gray Code Order

0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

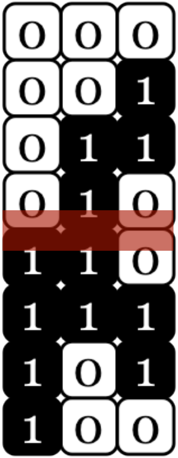
Volume Control Application

Boom Box Volume

Lexicographic Order



Gray Code Order

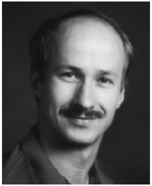


Modern Historic Milestones

1953 **Pulse code communication** Frank Gray patents BRGC

1978 **Combinatorial algorithms** A. Nijenhuis and H. Wilf

1995 **Combinatorial object server** **Frank Ruskey** $\longrightarrow$
`http://www.theory.csc.uvic.ca/~cos/`



1995 **Online encyclopedia of integer sequences** N. Sloane and S. Plouffe
`http://oeis.org/`

1996 **A survey of combinatorial Gray codes** Carla Savage

2005 **The art of computer programming: volume 4A** Donald Knuth

Combinatorial Generation Frameworks

ECO (Enum. Combinatorial Objects) Barucci, Del Lungo, Pergola, Pinzani (1999)

- a recursive method that defines an operator that performs local expansion
- Dyck and Motzkin paths, generalized Fibonacci sequences, polyominoes, ...

Family Tree Uno, Nakano, Uehara (2000)

- applied mostly to graph structures
- directed spanning trees, rooted trees, bipartite minimum edge colourings, ...

Reflectable Languages Li, S. (2004)

- generalizes languages that can apply reflection as per the BRGC
- restricted growth strings, k-ary trees, open meandric systems, ...

Bubble Languages (cool-lex) Ruskey, S., Williams (2009)

- the first 01 can be replaced by 10 to obtain a new object from the set
- k-ary trees, unit interval graphs, necklaces, Lyndon words, ...

Greedy Method Williams (2013)

- generalizes many previously published algorithms including the BRGC

But do these Algorithms have any Application?

Beckett Gray code

In my quintet, I actually used the Beckett-Gray code for $n = 5$ that appears in your article ... –Daniel Wolf, Composer

Unlabled necklaces

I work on symmetry breaking while traversing a search tree of a constraint satisfaction problem in constraint-programming fashion. –Pierre Flener

Fixed density bracelet

BTW I used your code to construct quantum error-correcting codes. My research area is quantum information/computation. –S. Yong Looi

Fixed content universal cycle

I wanted to try constructing the lookup table as a de Bruijn sequence, to reclaim some space. It may even lead to better cache usage. –Alex Bowe

Chord diagrams

We're playing with chord diagrams for a couple of different projects, and I came across your algorithm to quickly enumerate them. –Tom Boothby

Another Application: Universal Cycles

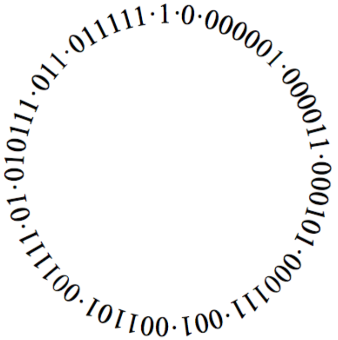
A **universal cycle** for a set S is a cyclic sequence $u_1 u_2 \cdots u_{|S|}$ where each substring of length n corresponds to a unique object in S

When S is the set of k-ary strings of length n, universal cycles are commonly called **de Bruijn sequences**

Necklaces N(6)	Aperiodic prefixes
000000	0
000001	000001
000011	000011
000101	000101
000111	000111
001001	001
001101	001101
001111	001111
010101	01
010111	010111
011011	011
011111	011111
111111	1

lexicographic order ↓

de Bruijn sequence FKM(6)



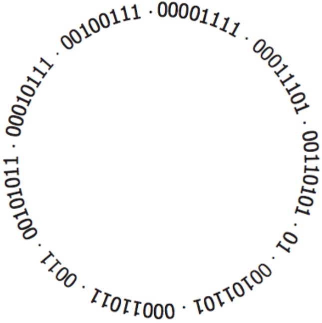
A Universal Cycle for d-Subsets

Using **cool-lex** order for necklaces with weight d and length n, we can construct a universal cycle for the subsets of size d from a set of size n.

cool-lex order (↑)

necklaces $\mathbf{N}_4(8)$	aperiodic prefixes
00001111	00001111
00011101	00011101
00110101	00110101
01010101	01
00101101	00101101
00011011	00011011
00110011	0011
00101011	00101011
00010111	00010111
00100111	00100111

fixed-weight de Bruijn sequence $\mathbb{C}_4(8)$



Each string of length $n - 1$ in the universal cycle corresponds to a binary string of length n and weight d (the last bit is redundant).

Coming Up ...

Aaron Williams	**Iterative Gray codes**
Ryuhei Uehara	On generation of graphs with geometric representations
Xi Sisi Shen	A *Hot Potato* transposition Gray code for permutations
~~*Victoria Horan*~~	~~*Universal cycles for weak orders*~~

Enjoy the rest of the talks ... *and long may your big jib draw!*