Discrete Applied Mathematics 210 (2016) 61–74

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Greedy flipping of pancakes and burnt pancakes

Joe Sawada [a,*], Aaron Williams [b]

[a] School of Computer Science, University of Guelph, Canada
[b] Department of Mathematics and Statistics, McGill University, Canada

ARTICLE INFO

Article history:
Received 5 December 2013
Received in revised form 16 November 2015
Accepted 5 February 2016
Available online 6 May 2016

Keywords:
Greedy algorithm
Gray code
Permutations
Signed permutations
Prefix-reversal
Symmetric group
Signed symmetric group
Cayley graph
Hamilton cycle

ABSTRACT

We prove that a stack of n pancakes is rearranged in all $n!$ ways by repeatedly applying the following rule: *Flip the maximum number of pancakes that gives a new stack*. This complements the previously known pancake flipping Gray code (Zaks, 1984) which we also describe as a greedy algorithm: *Flip the minimum number of pancakes that gives a new stack*. Surprisingly, these maximum and minimum flip algorithms also rearrange stacks of n 'burnt' pancakes in all $2^n n!$ ways. We conjecture that these four algorithms are essentially the only greedy algorithms for rearranging pancakes and burnt pancakes in all possible ways using flips.

1. Introduction

Take a stack of n distinct pancakes, numbered $1, 2, \ldots, n$ by increasing diameter, and repeat the following: *Flip the maximum number of topmost pancakes that gives a new stack*. For example, if the first stack is 12345 when read from top to bottom, then the second stack is created by flipping all five pancakes to give 54321. To create the third stack from the second stack, we cannot flip all five pancakes (since it would recreate 12345), however we can flip the top four pancakes to give 23451. This process is a greedy algorithm, and Fig. 1 illustrates the resulting list of stacks.

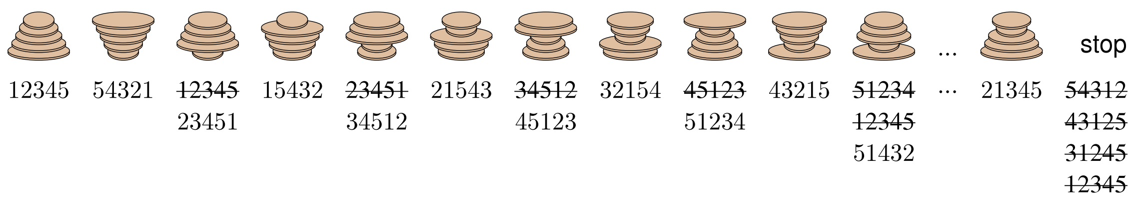

Fig. 1. Greedily flipping the maximum number of topmost pancakes from 12345. The order is read from left-to-right, and previously created stacks that are rejected by the algorithm are crossed out. All $5! = 120$ stacks are created. The last stack is 21345 since each flip gives a previous stack. In particular, flipping the top two pancakes gives the first stack 12345.

* Corresponding author.
E-mail addresses: jsawada@uoguelph.ca (J. Sawada), haron@uvic.ca (A. Williams).

http://dx.doi.org/10.1016/j.dam.2016.02.005

Formally, each stack of pancakes is a permutation of $\{1, 2, \ldots, n\}$ in one-line notation, and flipping the topmost k pancakes corresponds to a *prefix-reversal* of length k in the permutation. When using the pancake flipping metaphor, the reader can visualize a spatula being used for each flip. The same metaphor can be applied to 'burnt' pancakes that have two distinct sides; the 'burnt' side of each pancake alternates facing up and down when it is flipped. In this case, a stack of burnt pancakes is formalized as a *signed permutation* of $\{1, 2, \ldots, n\}$ and flipping the topmost k pancakes corresponds to a *complemented prefix-reversal* of length k in the signed permutation. Overlines are used to represent negative elements in a signed permutation. For example, applying a complemented prefix-reversal of length three to the signed permutation $\bar{1}3\bar{4}\bar{2}5$ results in $4\bar{3}1\bar{2}5$. The greedy algorithm that flips the maximum number of pancakes can also be applied to stacks of burnt pancakes, as illustrated in Fig. 2.

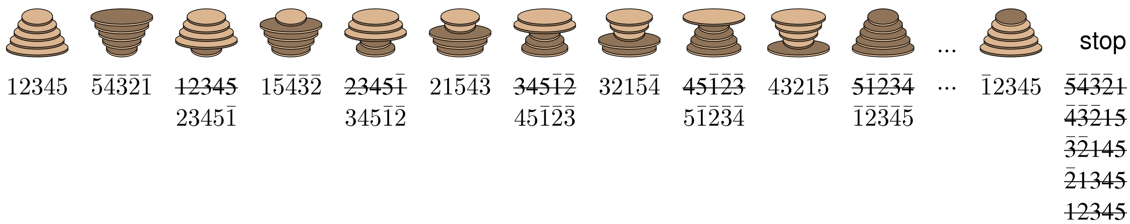

Fig. 2. Greedily flipping the maximum number of topmost burnt pancakes starting from 12345. All $2^5 5! = 3840$ stacks are created. The last stack is $\bar{1}2345$ since each flip gives a previous stack. In particular, flipping the top pancake gives the first stack 12345.

Amazingly, the lists generated by the greedy algorithm are both exhaustive for $n = 5$. In other words, the greedy algorithm generates all $5! = 120$ permutations and all $2^5 5! = 3840$ signed permutations before it terminates. We will prove that this result holds for all $n \geq 1$. Furthermore, we prove that the analogous *minimum flip* greedy algorithm also creates all $n!$ permutations and $2^n n!$ signed permutations. Collectively, these four results form the basis for this article.

To understand the significance of these results, let us consider two similar greedy algorithms. A *prefix-rotation of length j* moves the jth symbol to the beginning and the first $j - 1$ symbols are moved one position to the right. For example, 54321 becomes 25431 after a prefix-rotation of length four. A metaphor for this scenario is a vertical column of n distinct balls, where prefix-rotations of length j are performed by grabbing the jth ball and dropping it at the top of the column. Fig. 3 shows the result of greedily rotating the maximum length prefix of the permutation representing each container starting from 1234. Similarly, Fig. 4 shows the result of greedily rotating the minimum length prefix starting from 1234. In both cases the algorithm terminates before all $4! = 24$ permutations are created.

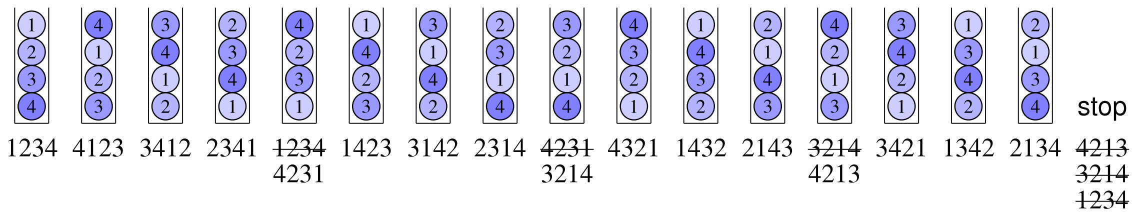

Fig. 3. Greedily rotating the maximum length prefix starting from 1234 terminates after creating only 16 permutations.

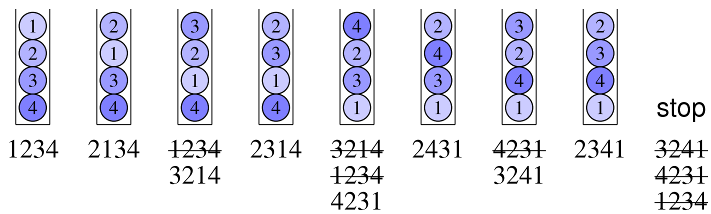

Fig. 4. Greedily rotating the minimum length prefix starting from 1234 terminates after creating only 8 permutations.

Readers are likely familiar with the *binary reflected Gray code* [6], which orders the 2^n n-bit binary strings so that successive strings differ by a single bit complementation. In general, the term *Gray code* can be used for any exhaustive ordering of a set of combinatorial objects in which successive objects are "close to each other" according to some measure or operation. For surveys on Gray codes of permutations and other objects see Sedgewick [14], Savage [11], and Section 7.2.1.2 of Knuth [10]. We describe our main results using the language of Gray codes as follows: