COMPUTING MAXIMUM NUMBER OF FLIPS FOR n-PANCAKE PROBLEM

27th April,2016

Nayan Chandni Sarat C. Lingamarla

Abstract

Computing the maximum number of flips or prefix reversals it takes to sort a given permutation of numbers or pancakes into a sorted stack using pancake sorting algorithm.It is also equivalent to finding the diameter of a pancake graph i.e. a graph with different permutations of the stack as its nodes and edges representing 1 prefix reversal between the nodes it connects. Finding the maximum number of flips to sort a stack of pancakes has important applications in finding latency in networks with pancake graph as a topology.

1. Introduction

The pancake problem is given as "The chef in our place is sloppy, and when he prepares a stack pancakes they come out all different sizes. Therefore, when I deliver them to a customer, on the way to the table I rearrange them (so that the smallest winds up on top, and so on, down to the largest on the bottom) by grabbing several ones from the top and flipping them over, repeating this (varying the number I flip) as many times as necessary. If there are n pancakes, what is the maximum number of flips (as a function) that I will ever have to use to rearrange them?" [1].

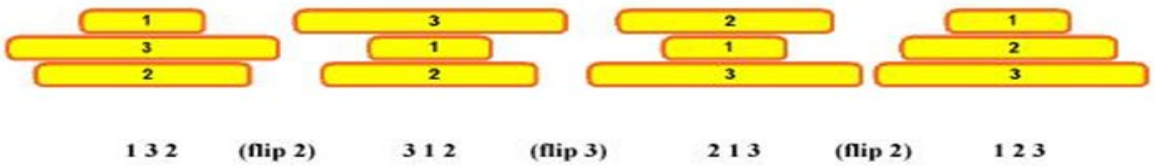

Fig 1 : An example of pancake sorting method applied on stack of 3 arranged as permutation (1 3 2)

(Source : website : http://poj.org/problem?id=2275)

Let S_n be the set of all permutations of a n pancake stack and **Π** be one of the permutations in the set.If **Π** = xy where x and y are substrings in the permutation,then let $\boldsymbol{\sigma} = x^R y$ denotes the

prefix reversal or flip at the end of x substring and e^n is the sorted stack of pancakes obtained after performing several prefix reversals.If $\mathbf{\Pi}^x = e^n$,then $\boldsymbol{x}$ is the sequence of prefix reversals that need to performed on $\mathbf{\Pi}$ to get the sorted stack.[5]

The problem then is to find the length of the sorting sequence $\boldsymbol{x}$. Similarly,the sorting sequences for all the permutations in the stack are calculated.The maximum of such lengths is calculated to find the maximum number of flips or worst case number of flips needed to sort a stack of pancakes. However,as the number of pancakes increases,the number of permutations also increases because there would be n! possible permutations for a n-pancake stack,thereby increasing the complexity of the problem.

2. Background

The problem was first posed by Jacob Goodman, writing under the name “Harry Dweighter” , in 1975 in the ”American Mathematical Monthly” (vol. 82, p. 1010, 1975). In general terms, it concerns the number of flips, or ”prefix reversals”, needed to sort the elements of an arbitrary permutation. Initial work on the problem established the limits for Pn that we saw above.

In 1979, William H. “Bill” Gates and Christos H. Papadimitriou improved the bounds of n and $2n - 3$, showing that flips always suffice and that flips may be needed. They showed the bounds $17n/16 \leq Pn \leq 5/3(n + 1)$.

The paper was based on research conducted when Bill Gates was an undergraduate at Harvard University before he went on to found Microsoft, though it was only published a few years later.The paper proposes a method where adjacencies are defined in the permutations of pancakes.There is supposed to exist an adjacency when two pancakes are adjacent to each other and there exists no intermediate size between them.Elements in the pancake stack are said to be free when they have no adjacencies and belong to a block if they have adjacent elements.The algorithm defines cases depending on whether an element is free or belongs to a block and its relative position within a block [6].

In 1997, Mohammad H. Heydari and I. Hal Sudborough improved the lower bound, and worked out the pancake numbers up to 13. The upper bounds were improved by Chitturi and others, and the best current bounds are $15n/14 \leq Pn \leq 18n/11$.

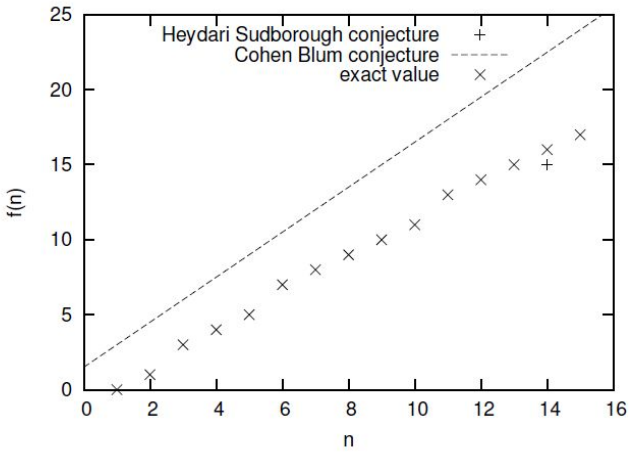

Figure 1 : Exact and conjectured values of Pn

A variation of the Pancake Problem was proposed by Bill Gates and Papadimitriou in 1979 in which the pancakes had one burnt side that cannot face upwards in the end, alongwith being sorted in an increasing order of size.The representation of the pancakes now being numbered as integers in increasing value was changed to using signed integer wherein a negative integer value indicated the burnt side being up.

Later, David S. Cohen , gave upper and lower bounds.The lower bound and upper bound for a stack n>=10 were found to be 3n/2 and 2n-2.

3. Resources and Methods

3.1. Approach

For the prefix reversal problem in consideration over here,the simplest approach suggested has been as follows:

1) Select the largest integer from the stack
2) Flip the stack starting from the starting point upto the largest integer(This brings the largest pancake on the top)

3) Now flip the entire stack,fixing the largest at the bottom.Now repeat process above two process recursively with all but the last pancake.

Since every pancake has a different size, let's rank the pancakes from 1 (smallest) to n (largest) and denote pancake i with the number. Example,for n=5 a sorted stack would be(1 2 3 4 5).

Now,starting with counting the number of flips for a pancake of n size,1 flip would be sufficient to order them only in the case (n…..i+2;i+1;i;i-1;i-2;.....1) and we also observe that the only stacks that can be sorted using one flip looks like: (i ; i-1 ; i-2 ; …..1 ; i+1 ; i+2 ;.....n) for some i□ (2,3…..n).

3.2. The n^{th} Pancake number

The maximum number of flips ever needed to rearrange a stack of n pancakes is now known as the nth pancake number, P_n. In technical terms, the problem concerns the number of flips, or prefix reversals, needed to sort the elements of an arbitrary permutation.

P_n = Max (Minimum number of flips required to sort S)

all starting n-pancake stacks S

P_0: 0. Any empty stack of pancakes is already sorted.

P_1: 0. So is any stack of a single pancake.

P_2: 1. With two pancakes in the stack, either the stack is already in the correct order or it is upside-down. At most, one flip would be required.

3.2. Theorem statement

Initial work established that Pn had to be at least n and no more than 2n – 3, for n greater than 2.

Proof : if the number of flips required by this modified “bring-to-top" is denoted by T(n), then it Satisfies:

$$T(1) = 0$$

$$T(2) <= 1$$

T(n) <=2 + T(n -1) for n >= 3 which solves to T(n) 2n <= 3 for n>= 2.

3.3. Implementation

The clojure code for this algorithm is provided in Appendix-I.It uses functional programming paradigms to the fullest by describing functions of steps such as above-spatula and under-spatula while flipping pancakes in a stack.A map function applies the sorting method to all permutations of a stack S and finds the maximum of the flips required,as mentioned in the nth Pancake number definition.

4. Results

In theory and through experimentation,the Pancake numbers upto P(17) are known while an ambiguity exists for the value of P(18) which is an open-question till date[3].The difficulty lies in searching through all the existing permutations of such a number to find the suitable stack S which would be the most challenging to sort.Mathematically,sequences corresponding to 18 pancakes produce a total of $18! = 6.40237371 \times 10^{15}$.The amount of data produced during computation may take several days for processing.[4]*

Through experiment using the mentioned clojure code,the following values have been established.The tabular data of observations vs expected values have been given in Table 1 below.Here the program has been run on a standalone machine with 8GB RAM capacity.

(*The speed can be greatly increased using distributed computing techniques.)

Table 1.Maximum number of flips n taken to sort the worst case of a stack S of pancakes using k flips.Estimated with the running time of the program.

Number n	Flips k (actual)	Flips k (experiment)	Stack	Time spent by program
3	3	3	6	0.642558 msecs
4	4	4	24	2.318929 msecs
5	5	5	120	11.185469 msecs
6	7	7	720	37.061041 msecs
7	8	8	1440	314.238903 msecs
8	9	9	40320	2813.801357 msecs
9	10	10	362880	33444.891638 msecs
10	11	11	3628800	339350.620265 msecs
11	13	13	39916800	3958822.064593 msecs
12	14	14	479001600	5.8761635361204E7 ms
13	15	-	6227020800	-
14	16	-	87178291200	-
15	17	-	1.3076744e+12	-
16	18	-	2.092279e+13	-
17	19	-	3.5568743e+14	-

(Sources for actual flips:The Online Encyclopedia of Integer Sequences of Neil Sloane,On average and highest number of flips in pancake sorting[2])

Appendix-I

```
;; This function finds the largest element from the list[9]
(defn biggest [coll] (apply max coll))
;; This function returns index of largest element from list[9]
(defn index-of-biggest [coll] (.indexOf coll (biggest coll)))

;;Pancake sorting that applies sort function to all elements in list*[9]
(defn pancake-sort [coll]
  (if (> 2 (count coll))
          coll
          (let [spatula       (inc (index-of-biggest coll))
          above-spatula (take spatula coll)
          under-spatula (drop spatula coll)
          biggest-under (reverse (concat (reverse above-spatula) under-spatula))]
           (
if   (  not  (   apply <= coll  )   )
(dosync (alter x inc))
)
 (conj (vec (pancake-sort (butlast biggest-under)))
          (last biggest-under))
           )))

;; Counts the number of flips required for sorting to final state
(defn count-flips [coll]
(def x (ref 0))
(pancake-sort coll)@x)

;;Produces all permutations for a given number n
(defn numx [x] ( range 1 (+ x 1)) )
;;Applies counting function to all permutations and times it
 (time(apply max (map  count-flips (permutations (numx 3)))))
```

*Note: The code from reference [9] was modified here.

References

[1] H. Dweighter, Problem e2569. Amer. Math. Monthly,82:1010, 1975.

[2]On average and highest number of flips in pancake sorting ,Josef Cibulka

[3]Sloane, N. J. A. ”My Favorite Integer Sequences.” In Sequences and Their Applications (Proceedings of SETA ’98) (Ed. C. Ding, T. Helleseth, and H. Niederreiter). London Springer-Verlag, pp. 103-130, 1999.http://www.research.att.com/ njas/doc/sg.pdf.

[4]Additional information about pancake sorting can be found at http://mathworld.wolfram.com /PancakeSorting.html

[5]Yuusuke Kounoike,Keiichi Kaneko andYuji Shinano.Computing the Diameters of 14- and 15-Pancake Graphs.IEEE,Proceedings of the 8th International Symposium on Parallel Architectures, Algorithms and Networks,2005

[6]William H.Gates,Christos H.Papadimitriou.Bounds for Sorting by Prefix Reversal,Discrete Math.,27:47-57,1979

[7]Cohen, D.S., and M. Blum. 1995. On the problem of sorting burnt pancakes. Discrete Applied Mathematics 61:105-120.

[8]Heydari, M.H., and I.H. Sudborough. 1997. On the diameter of the pancake network. Journal of Algorithms.

[9]https://github.com/pokle/coding-exercise-clojure-pancake-sort