ELSEVIER Discrete Applied Mathematics 61 (1995) 105–120

DISCRETE APPLIED MATHEMATICS

On the problem of sorting burnt pancakes

David S. Cohen*[1], Manuel Blum[2]

Computer Science Division, University of California, Berkeley, CA 94720, USA

Received 30 June 1992; revised 5 October 1993

Abstract

The "pancake problem" is a well-known open combinatorial problem that recently has been shown to have applications to parallel processing. Given a stack of n pancakes in arbitrary order, all of different sizes, the goal is to sort them into the size-ordered configuration having the largest pancake on the bottom and the smallest on top. The allowed sorting operation is a "spatula flip", in which a spatula is inserted beneath any pancake, and all pancakes above the spatula are lifted and replaced in reverse order. The problem is to bound $f(n)$, the minimum number of flips required in the worst case to sort a stack of n pancakes. Equivalently, we seek bounds on the number of *prefix reversals* necessary to sort a list of n elements. Bounds of $17n/16$ and $(5n + 5)/3$ were shown by Gates and Papadimitriou in 1979. In this paper, we consider a traditional variation of the problem in which the pancakes are two sided (one side is "burnt"), and must be sorted to the size-ordered configuration in which every pancake has its burnt side down. Let $g(n)$ be the number of flips required to sort n "burnt pancakes". We find that $3n/2 \leqslant g(n) \leqslant 2n - 2$, where the upper bound holds for $n \geqslant 10$. We consider the conjecture that the most difficult case for sorting n burnt pancakes is $-I_n$, the configuration having the pancakes in proper size order, but in which each individual pancake is upside down. We present an algorithm for sorting $-I_n$ in $23n/14 + c$ flips, where c is a small constant, thereby establishing a bound of $g(n) \leqslant 23n/14 + c$ under the conjecture. Furthermore, the longstanding upper bound of $f(n)$ is also improved to $23n/14 + c$ under the conjecture.

1. Introduction

The "pancake problem" was posed as follows in [2]:

> The chef in our place is sloppy, and when he prepares a stack of 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

*Correspondence address: 155 S. Rexford Dr #F, Beverly Hills, CA 90212, USA. E-mail: dcohen@melody.berkeley.edu.

[1] Supported in part by an ONR Graduate Fellowship and in part by NSF grant no. CCR-9201092.

[2] Supported in part by NSF grant no. CCR-9201092.

SSDI 0166-218X(94)00009-3

on, down to the largest at the bottom) by grabbing several 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 $f(n)$ of n) that I will ever have to use to rearrange them?

In [3], Gates and Papadimitriou derive bounds of $17n/16 \leqslant f(n) \leqslant (5n + 5)/3$, where the lower bound holds for n a multiple of 16.

Herein we consider a traditional variation on the "pancake problem", known as the "burnt pancake problem", in which the pancakes are two-sided (one side is burnt). Initially, the pancakes are arbitrarily ordered and each pancake may have either side up. After sorting, the pancakes must not only be in size order, but must have their burnt sides face down. Let $g(n)$ be the number of flips required to sort n burnt pancakes in the worst case. Clearly, $g(n) \geqslant f(n)$, since any algorithm for sorting burnt pancakes works for unburnt pancakes as well if we simply ignore sidedness.

A further specialization of the burnt pancake problem in which all of the pancakes begin, as well as end, with the burnt side down is considered in [3]. For that case, they find bounds of $3n/2 - 1$ and $2n + 3$.

In this paper we obtain bounds of $3n/2 \leqslant g(n) \leqslant 2n - 2$, where the upper bound holds for $n \geqslant 10$. We then consider a conjecture that the worst case for sorting n burnt pancakes is $-I_n$, the configuration having all pancakes in proper size order, but in which each individual pancake is upside down. We present an algorithm for sorting $-I_n$ in $23n/14 + c$ flips, where c is a small constant. This gives us an upper bound, under the conjecture, of $g(n) \leqslant 23n/14 + c$. Since $g(n) \geqslant f(n)$, the longstanding upper bound on $f(n)$ is also improved to $23n/14 + c$ under the conjecture.

Interestingly, the pancake problem may have practical applications in parallel processing. The pancake graph (on burnt or unburnt pancakes) described in Section 4 looks promising as a network for parallel algorithms; in particular, it has sublogarithmic diameter and degree as a function of the number of processors (vertices) in the network [1]. In these respects it is better than, for example, the hypercube, which has logarithmic diameter and degree. Routing between processors on the pancake graph is equivalent to sorting stacks of pancakes, and so is not too difficult. Pancake network algorithms are known for broadcast, parallel prefix, and binary tree simulation [1], as well as sorting [6], and hypercube simulation [5].

2. Notation

We represent a stack of n burnt pancakes as an n-dimensional vector containing the integers 1 through n in some order. 1 represents the smallest pancake, n the largest. A minus sign preceding a number indicates that the pancake is upside-down. The topmost number represents the pancake currently atop the stack. The symbol $\vdots$ represents one or more pancakes. The symbol $\vdash$ between two vectors indicates that the first can be transformed into the second with one flip. Similarly, $\overset{m}{\vdash}$ indicates that the first can be transformed into the second with m flips.

In general, we denote an arbitrary stack of pancakes by a capital letter such as A or B. The stack obtained from A by leaving its pancakes in the same order but flipping each pancake in place is denoted by $-A$. We use I or I_n to denote the sorted configuration having 1 on top and n on bottom.

3. A $2n$ upper bound

We first note that there is a trivial upper bound of $3n$: flip the largest pancake to the top, flip this pancake again if necessary, then flip the entire stack, bringing the largest pancake (in a rightside up position) to the bottom. Repeat this procedure recursively on the $n - 1$ remaining pancakes. This sorts the stack in at most $3n$ flips. (Note that the smallest pancake will in fact require at most 1 flip; hence this algorithm actually gives an upper bound of $3n - 2$.) We now present an algorithm which requires only $2n$ flips in the worst case.

When a flip brings together two consecutive pancakes p and $p + 1$ in the correct order (that is, with the burnt side of p touching the unburnt side of $p + 1$), we say that p and $p + 1$ have been *joined*. Note that in the figure below, (a) and (b) represent stacks in which p and $p + 1$ are joined, whereas (c) and (d) represent stacks in which p and $p + 1$ are not joined.

$$\underset{\text{(a)}}{\begin{bmatrix} \vdots \\ p \\ p+1 \\ \vdots \end{bmatrix}} \quad \underset{\text{(b)}}{\begin{bmatrix} \vdots \\ -(p+1) \\ -p \\ \vdots \end{bmatrix}} \quad \underset{\text{(c)}}{\begin{bmatrix} \vdots \\ p+1 \\ p \\ \vdots \end{bmatrix}} \quad \underset{\text{(d)}}{\begin{bmatrix} \vdots \\ -p \\ -(p+1) \\ \vdots \end{bmatrix}}$$

Once p and $p + 1$ are joined, we may choose never again to separate them. In this case the join is equivalent to reducing number of pancakes in the stack by one. (Placing the largest pancake n rightside up at the bottom of the stack is also considered a join, since thereafter we need never again move it.) After n joins, the stack has been sorted.

This view of the problem leads to a sorting algorithm requiring $2n$ flips in the worst case. For nearly every configuration of pancakes, the algorithm proceeds by producing one join in at most two flips. In the analysis one special configuration arises from which no join is possible within two flips; however, a second algorithm requiring exactly $2n$ flips is presented for this special case. We conclude that any stack of burnt pancakes can be sorted in at most $2n$ flips.

Burnt pancake sorting algorithm. Given any stack of burnt pancakes, find the case below which describes that stack and perform the prescribed flips, reducing the number of pancakes by one. Repeat until no pancakes remain (the stack is sorted).

Case 1: At least one pancake is rightside up in the stack. Let p be the largest such pancake. Note that $p+1$ must therefore be upside down, unless $p = n$.

(a) $-(p+1)$ is lower than p in the stack:

$$\begin{bmatrix} \vdots \\ p \\ \vdots \\ -(p+1) \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} p+1 \\ \vdots \\ -p \\ \vdots \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} \vdots \\ -(p+1) \\ -p \\ \vdots \\ \vdots \end{bmatrix}$$

(b) $-(p+1)$ is higher than p in the stack:

$$\begin{bmatrix} \vdots \\ -(p+1) \\ \vdots \\ p \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} -p \\ \vdots \\ p+1 \\ \vdots \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} \vdots \\ p \\ p+1 \\ \vdots \\ \vdots \end{bmatrix}$$

(c) p is the largest pancake in the stack; i.e., $p = n$.

$$\begin{bmatrix} \vdots \\ \vdots \\ n \\ \vdots \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} -n \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \vdots \\ n \end{bmatrix}$$

We see that if any pancake is rightside up in the stack, then we can achieve a join within two flips using (a), (b), or (c).

Case 2: All pancakes are upside down.

(a) For some p, $-(p+1)$ is higher than $-p$ in the stack.

$$\begin{bmatrix} \vdots \\ -(p+1) \\ \vdots \\ -p \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} p+1 \\ \vdots \\ \vdots \\ -p \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} \vdots \\ \vdots \\ -(p+1) \\ -p \\ \vdots \end{bmatrix}$$

(b) For all p, $-(p+1)$ is below $-p$. The pancakes must be in exactly the configuration $-I_n$, and it is not hard to see that no join is possible within two flips. Before dealing with this case, we summarize Cases 1 and 2(a) in the following theorem.

Theorem 1. *If a stack of n burnt pancakes is not in the configuration* $-I_n$, *then a join can be achieved within two flips.*

We now present an extremely simple algorithm for sorting $-I_n$ in $2n$ flips:

Algorithm (*).
(1) Flip all n pancakes.
(2) Flip the top $n-1$ pancakes.
(3) Repeat (1) and (2) n times.

The proof that the algorithm sorts the stack $-I_n$ in $2n$ flips uses the following claim.

Claim. *After running algorithm (*) for $2i$ flips, $i \leqslant n$, on the stack $-I_n$, the stack will be in the following configuration:*

$$\begin{bmatrix} -(i+1) \\ -(i+2) \\ \vdots \\ -(n-1) \\ -n \\ 1 \\ 2 \\ \vdots \\ i-1 \\ i \end{bmatrix}$$

Proof (*by induction*). For $i = 0$, the claim is trivially true. Now assume the claim is true for i. Executing the next two flips called for by the algorithm,

$$\begin{bmatrix} -(i+1) \\ -(i+2) \\ \vdots \\ -(n-1) \\ -n \\ 1 \\ 2 \\ \vdots \\ i-1 \\ i \end{bmatrix} \vdash \begin{bmatrix} -i \\ -(i-1) \\ \vdots \\ -2 \\ -1 \\ n \\ n-1 \\ \vdots \\ i+2 \\ i+1 \end{bmatrix} \vdash \begin{bmatrix} -(i+2) \\ -(i+3) \\ \vdots \\ -(n-1) \\ -n \\ 1 \\ 2 \\ \vdots \\ i \\ i+1 \end{bmatrix}$$

Thus the claim holds for $i+1$ and so for all $0 \leqslant i \leqslant n$.

The correctness of algorithm (*) follows immediately by noting that for $i = n$, the claim states that the stack will be sorted . □

Taken together, Cases 1, 2(a), and algorithm (*) give an algorithm for sorting any stack of n burnt pancakes. Cases 1 and 2(a) are applied i times until the stack is sorted or until the configuration $-I_{n-i}$ arises. Then algorithm (*) is used. Since each of Cases 1(a)–(c) and 2(a) requires 2 flips, the total number of flips used will be $2i + 2(n - i) = 2n$. Thus we obtain:

Theorem 2. $g(n) \leqslant 2n$.

In Section 5, we improve this to $g(n) \leqslant 2n - 2$ for $n \geqslant 10$.

4. The pancake group

It is very useful to view the pancake problem in a group-theoretic setting. Here we focus primarily on the burnt pancakes, but the unburnt pancakes can be treated analogously.

The group B_n of burnt pancakes consists of all "signed permutations" on n elements. An element of B_n may reverse or leave unchanged the sign of each coordinate of an n-dimensional vector, then permute these coordinates in an arbitrary fashion. B_n is generated by the elements $\{b_1, \ldots, b_n\}$, where generator b_i corresponds to a flip of the top i burnt pancakes. Note that $b_i = b_i^{-1}$. Similarly, the group S_n of unburnt pancakes (i.e., the *symmetric group* of permutations on n elements) is generated by $\{u_2, \ldots, u_n\}$, where u_i corresponds to a flip of the top i unburnt pancakes. Again, $u_i = u_i^{-1}$. Note that u_1 is the identity, whereas $b_i b_1 b_i$ serves to flip the ith pancake in place.

Consider the Cayley graph of the group B_n with generators b_j, which we refer to as the *pancake graph*. Each vertex of the graph can be identified with a unique stack of n burnt pancakes, and two vertices are connected by an edge whenever a single flip transforms one into the other. A path through the graph from A to B corresponds to a sequence of flips transforming stack A into stack B. The graph has $2^n n!$ vertices and is regular of degree n.

If we place a marker on some vertex A of the pancake graph, then an element g of the group moves the marker from A to some other vertex B. We write $g(A) = B$, and we say g *sorts* A if $g(A) = I_n$. The identity element 1_n satisfies $1_n(A) = A$. We also define the group element -1_n to be the element satisfying $-1_n(A) = -A$. Note that $-1_n^2 = 1_n$, or $-1_n = -1_n^{-1}$.

Each group element is really just a signed permutation, and as such does not specify what path through the graph (sequence of flips) is taken in moving from A to B. We can obtain one such path by expressing g as a product of generators; however, there are an infinite number of possible expressions for any $g \in B_n$ (including the generators themselves). The burnt pancake problem can be viewed as the problem of bounding the number of generators required to express any element of the group B_n. The set of elements of B_n requiring the longest product of generators in their minimum expressions are referred to as the "longest" or worst-case elements of B_n.

The *distance* $d(A, B)$ between two pancake stacks A and B is the number of generators in the shortest expression of the element g for which $g(A) = B$. Equivalently, $d(A, B)$ is the number of edges in the shortest path from A to B in the pancake graph. Note that since the pancake graph is undirected, $d(A, B) = d(B, A)$.

An important property of both the burnt and unburnt pancake graphs is that they are *vertex symmetric*. That is, given any two vertices A and B, there is an automorphism of the graph mapping A to B (in fact, vertex symmetry is a property of all Cayley graphs [1]). The automorphism in this case is given by $h(A) \mapsto h(B)$ for all $h \in B_n$. Intuitively, the graph "looks the same" when viewed from any vertex. This leads to the following theorem.

Theorem 3. *Let W be the set of longest elements in a (burnt or unburnt) pancake group. Then for each $w \in W$, it is also the case that $w^{-1} \in W$, i.e., W is closed under inverses.*

Proof. Let A be some stack at the maximum distance k from I. Let w be the group element that sorts A; $w(A) = I$. Thus the length of w is k. But then $w^{-1}(I) = A$, and since distance is symmetric, the length of w^{-1} is also k. By the vertex symmetry property, we have that for all vertices B, both $w(B)$ and $w^{-1}(B)$ are at maximum distance k from B. □

Corollary 3.1. *If the longest element w of a (burnt or unburnt) pancake group is unique, then it is an involution.*

As an example, consider the six worst cases (located by computer search) for $n = 11$ unburnt pancakes, shown in Fig. 1. Note that (a)–(d) are involutions, while (e) and (f) are a pair of inverses.

$$\underset{(a)}{\begin{bmatrix}1\\5\\3\\10\\2\\6\\9\\11\\7\\4\\8\end{bmatrix}} \quad \underset{(b)}{\begin{bmatrix}1\\11\\3\\6\\9\\4\\7\\10\\5\\8\\2\end{bmatrix}} \quad \underset{(c)}{\begin{bmatrix}1\\11\\3\\8\\5\\10\\7\\4\\9\\6\\2\end{bmatrix}} \quad \underset{(d)}{\begin{bmatrix}7\\2\\9\\6\\11\\4\\1\\10\\3\\8\\5\end{bmatrix}} \quad \underset{(e)}{\begin{bmatrix}1\\3\\11\\5\\8\\6\\2\\9\\7\\10\\4\end{bmatrix}} \quad \underset{(f)}{\begin{bmatrix}1\\7\\2\\11\\4\\6\\9\\5\\8\\10\\3\end{bmatrix}}$$

Fig. 1. Worst cases for 11 unburnt pancakes.

Recall that a permutation π can be assigned a sign of $+1$ or -1 corresponding to whether its expression as a product of *transpositions* (which swap the positions of two items while leaving the rest unchanged) is of even or odd length, respectively. Similarly, the sign of a "signed permutation" σ is the sign of the corresponding unsigned permutation (i.e., that obtained by ignoring minus signs), times -1 for each minus sign in σ.

Let S^{U}_{-1} be the set of generators of the unburnt pancake group with sign -1 and let S^{B}_{-1} be the set of generators of the burnt pancake group with sign -1. It is not hard to show that

$$S^{\mathrm{B}}_{-1} = \{b_1, b_2, b_5, b_6, \dots, b_{4k+1}, b_{4k+2}, \dots\},$$

$$S^{\mathrm{U}}_{-1} = \{u_2, u_3, u_6, u_7, \dots, u_{4k+2}, u_{4k+3}, \dots\}.$$

Now for any stack A of pancakes, we can take the sign of the permutation which sorts it to obtain the parity of the number of generators from S^{B}_{-1} or S^{U}_{-1} in any sorting sequence for A.

5. Properties of $-I_n$

The remainder of this paper will be devoted to an analysis of the pancake problem under a conjecture that $-I_n$ is the worst case (i.e., requires the most flips to sort) among all stacks of n burnt pancakes.

Exhaustive computer search reveals that the conjecture is true for $n \leqslant 8$ (using information from the computer search, we shall demonstrate below that the conjecture also holds for $n = 9$ and $n = 10$). Unfortunately, computational tests of the conjecture rapidly become intractable beyond this point.

It is interesting to note that $-I_n$ proved to be the one difficult case for the algorithm of Section 3. Recall also Corollary 3.1, which stated that if the worst case for a particular value of n is unique, then the signed permutation which sorts it must be an involution. We have already observed that -1_n, which sorts $-I_n$, is indeed an involution (for $6 \leqslant n \leqslant 9$, $-I_n$ is in fact the unique stack at the maximum distance from I_n).

In fairness, we should also point out that for some n, including $n = 11, 14, 15$, we have found a stack $-J_n$ (the same as $-I_n$, except that the topmost pancake is rightside up) that requires exactly as many flips to sort as $-I_n$. $-J_n$ does not appear to be a candidate for a general worst case, however, since for some values of n, including $n = 6, 7, 8, 9, 10, 12$, $-J_n$ is strictly easier to sort than $-I_n$, whereas no value of n has been found for which $-J_n$ is strictly more difficult to sort than $-I_n$.

We now present a number of interesting properties of $-I_n$ and -1_n:

Theorem 4. *1_n and -1_n are the only elements of B_n that commute with every element of B_n; i.e., the center of B_n is precisely $\{1_n, -1_n\}$.*

Proof. Write $g[i]=j$ or $g[-i]=-j$ if the signed permutation represented by g moves the pancake at position i to position j without flipping it. Write $g[i]=-j$ or $g[-i]=j$ if g flips the pancake at position i and then moves it to position j.

Consider an arbitrary group element g, and arbitrary indices a and b, $-n \leqslant a, b \leqslant n$, such that $g[a]=b$ (equivalently, $g[-a]=-b$). Note that for all indices i, $-1_n[i]=-i$. Thus $g[-1_n[a]]=g[-a]=-b$. However, $-1_n[g[a]]=-1_n[b]=-b$ as well. Since g and the indices a, b were arbitrary, the result holds for all indices of all elements $g \in B_n$. Then $-1_n g = g(-1_n)$, so -1_n commutes with all $g \in B_n$.

Now consider any other element $h \in B_n$, $h \neq 1_n, -1_n$. Any such h must fall into one of the following two categories:

(1) $h[a]=\pm b$, $b \neq a$, for some $1 \leqslant a, b \leqslant n$. Then choose an element g such that $g[a]=a$, and $g[b]=c$, $c \neq b$, $c \geqslant 1$. (Such a g must exist for $n \geqslant 3$.) Now, $h[g[a]]=h[a]=\pm b$, but $g[h[a]]=g[\pm b]=\pm c \neq \pm b$. So $hg \neq gh$ and h is not in the center of B_n.

(2) $h[a]=\pm a$ for all $1 \leqslant a \leqslant n$. Then since $h \neq 1_n, -1_n$, there must be distinct indices a and b such that $h[a]=a$ and $h[b]=-b$. Choose any group element g such that $g[a]=b$ and $g[b]=a$. Then $h[g[a]]=h[b]=-b$, but $g[h[a]]=g[a]=b$. So $hg \neq gh$ and h is not in the center of B_n.

The identity 1_n is trivially in the center of B_n. Thus -1_n and 1_n are the only elements in the center of B_n. □

This result leads immediately to the following theorem.

Theorem 5. *If $b_{i_1} b_{i_2} \cdots b_{i_k} = -1_n$ is a factoring of -1_n into a sequence of k flips, then any cyclic permutation of the flip sequence, and any cyclic permutation of the reversal of the sequence, is also a factoring of -1_n.*

Proof. To cycle the factors, we repeatedly multiply both sides of the equation on the left and right by the first flip in the sequence. For example, we can cycle once by:

$$b_{i_1} b_{i_2} \cdots b_{i_k} = -1_n,$$

$$b_{i_1} b_{i_1} b_{i_2} \cdots b_{i_k} b_{i_1} = b_{i_1}(-1_n) b_{i_1},$$

$$b_{i_2} \cdots b_{i_k} b_{i_1} = (-1_n) b_{i_1} b_{i_1},$$

$$b_{i_2} \cdots b_{i_k} b_{i_1} = -1_n,$$

where we have made use of the facts that $b_j(-1_n) = (-1_n) b_j$ and $b_j = b_j^{-1}$. We could cycle again by multiplying on the left and right by b_{i_2}, and so on.

Note that the reversal of any sequence of flips represents the inverse of the sequence. Thus $b_{i_k} \cdots b_{i_1} = -1_n^{-1} = -1_n$. But then any cyclic permutation of the reversed sequence is also a factoring of -1_n by the argument above. □

Theorem 6. *There exists an optimum (shortest) sequence of flips for sorting* $-I_n$ *that begins with the flip of the entire stack.*

Proof. The flip of the entire stack must occur at least twice in any sequence which sorts $-I_n$: once to get the upside-down pancake $-n$ off the bottom, and once to put it back rightside up. By the previous theorem, we can cycle any optimum sequence until one of these flips appears first. □

Let $T(n)$ be the minimum number of flips necessary to sort $-I_n$. Then Theorem 6 gives us the following corollary.

Corollary 6.1. *If* $-I_n$ *can be sorted optimally in* $T(n)$ *flips, then*

$$\begin{bmatrix} n \\ n-1 \\ \vdots \\ 2 \\ 1 \end{bmatrix} \overset{T(n)-1}{\vdash} \begin{bmatrix} 1 \\ 2 \\ \vdots \\ n-1 \\ n \end{bmatrix}$$

Under the conjecture, $g(n) = T(n)$, so it will suffice to obtain bounds on $T(n)$. The theorem below gives a bound on $T(n+1)$ when $T(n)$ is known.

Theorem 7. $T(n+1) \leqslant T(n) + 2$.

Proof. We first note the more obvious fact that $T(n+1) \leqslant T(n) + 3$; we can simply flip the largest pancake to the top, flip this pancake by itself if necessary, flip the entire stack, then sort the remaining n pancakes in $T(n)$ steps.

To obtain the improved result, let $(\uparrow)$ represent an arbitrary "substack" of pancakes, and let $(\downarrow)$ represent that same substack flipped upside down. Observe that at some point in the process of sorting $-I_n$, the two smallest pancakes 1 and 2 must be joined together and never again separated. This can occur only in one of the following two ways:

$$\underset{\text{(a)}}{\begin{bmatrix} -1 \\ (\uparrow) \\ 2 \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} (\downarrow) \\ 1 \\ 2 \\ \vdots \end{bmatrix}} \quad \text{or} \quad \underset{\text{(b)}}{\begin{bmatrix} 2 \\ (\uparrow) \\ -1 \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} (\downarrow) \\ -2 \\ -1 \\ \vdots \end{bmatrix}}$$

If case (a) occurs, we sort $-I_{n+1}$ as follows: Treat pancakes 2 and 3 as the single pancake 2, and apply the same sequence of flips used to sort $-I_n$ up to the point that

(a) is about to occur. Then execute this sequence of three flips:

$$\begin{bmatrix} -1 \\ (\uparrow) \\ 3 \\ 2 \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} -3 \\ (\downarrow) \\ 1 \\ 2 \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} -2 \\ -1 \\ (\uparrow) \\ 3 \\ \vdots \end{bmatrix} \vdash \begin{bmatrix} (\downarrow) \\ 1 \\ 2 \\ 3 \\ \vdots \end{bmatrix}$$

At the end of these three flips, we once again treat 2 and 3 as the single pancake 2. The configuration of the stack is exactly the same as after executing the single flip in (a), so we continue with the remainder of the sequence for sorting $-I_n$. In all only two additional flips have been inserted into the sequence, so $T(n+1) \leqslant T(n) + 2$.

If case (b) occurs, a similar argument applies except that now we treat pancakes 1 and 2 as the single pancake 1 up to the point that (b) occurs. Again, the insertion of two additional flips suffices.

It might seem at first that some complication could arise if pancakes 2 and 3 are already joined at the time that (a) or (b) occurs. However, a slightly more careful analysis shows that even after inserting the two extra flips needed to sort $-I_{n+1}$, the 2 and 3 remain joined.

We assumed the existence of pancakes 1 and 2 in $-I_n$, so we find that for $n \geqslant 2$, $T(n+1) \leqslant T(n) + 2$. □

Theorem 8. $g(n+1) \leqslant g(n) + 2$.

Proof. By Theorem 1, any stack of $n+1$ burnt pancakes other than $-I_{n+1}$ can be reduced to a stack of n burnt pancakes in two flips. But by Theorem 7, $-I_{n+1}$ requires at most two more flips than $-I_n$. Thus all stacks of $n+1$ pancakes require at most two more flips than some stack of n pancakes. □

Corollary 8.1. *$-I_9$ is the unique worst case for $n = 9$ burnt pancakes; thus $g(9) = T(9) = 17$.*

Proof. $T(9) = 17$ by computer search. But by Theorem 8, $g(9) \leqslant g(8) + 2 = 17$. Thus $-I_9$ is a worst case for $n = 9$; now we must show it is unique. Exhaustive computer search has shown that $-I_8$ is the unique worst case on 8 pancakes, requiring 15 flips. But by Theorem 1, any hypothetical stack of 9 burnt pancakes, other than $-I_9$, which requires 17 flips to sort can be reduced within two flips to a stack of 8 pancakes. This resulting stack of 8 pancakes must require at least 15 flips to sort, else we could have sorted the original stack of 9 pancakes in less than 17 flips. Since there is only one stack of 8 pancakes requiring 15 flips, any stack of 9 pancakes, other than $-I_9$, requiring 17 flips must be transformable to $-I_8$ in two flips. There are 72 such stacks;

computer search shows that all can be sorted in at most 16 flips. Hence $-I_9$ is the unique worst case on 9 burnt pancakes. □

We can now use this fact to verify the conjecture for the case $n = 10$.

Corollary 8.2. *$-I_{10}$ is a worst case for $n = 10$ burnt pancakes; thus $g(10) = T(10) = 18$.*

Proof. The same argument used in the proof of Corollary 8.1 applies once again. There are 90 stacks of 10 burnt pancakes that can be reduced in two flips to $-I_9$; these are the only stacks which could possibly require 19 flips to sort. Computer search shows that all can be sorted within 17 flips. Thus there are no stacks of 10 pancakes requiring 19 flips to sort, and since $-I_{10}$ requires 18 flips, it is a worst case. □

Corollary 8.3. *For $n \geqslant 10$, $g(n) \leqslant 2n - 2$.*

Proof. Follows immediately from Corollary 8.2 and Theorem 8. □

The next theorem is useful in pruning computer searches for an optimal sequence for sorting $-I_n$.

Theorem 9. *Call two consecutive flips f_i and f_{i+1} a useless pair if the number of items they flip differs by at most 1. If there is a sequence of k flips $(f_1, \ldots, f_k)$ for sorting $-I_n$ that contains a useless pair, then $T(n - 1) \leqslant k - 2$.*

Proof. The case $f_i = f_{i+1}$ is trivial, since then $f_i = f_{i+1}^{-1}$ and we can simply eliminate the useless pair of flips.

Now without loss of generality, let f_1 flip the top j pancakes and let f_2 flip the top $j + 1$ pancakes, where we have used Theorem 5 as necessary to reverse the sequence and/or cycle f_i to the first position. Then we have a sequence of the form:

$$\begin{bmatrix} -1 \\ \vdots \\ -j \\ -(j+1) \\ -(j+2) \\ \vdots \\ -n \end{bmatrix} \vdash \begin{bmatrix} j \\ \vdots \\ 1 \\ -(j+1) \\ -(j+2) \\ \vdots \\ -n \end{bmatrix} \vdash \begin{bmatrix} j+1 \\ -1 \\ \vdots \\ -j \\ -(j+2) \\ \vdots \\ -n \end{bmatrix} \overset{k-2}{\vdash} \begin{bmatrix} 1 \\ \vdots \\ j \\ j+1 \\ j+2 \\ \vdots \\ n \end{bmatrix}$$

But then we can sort $-I_{n-1}$ by using a variation on the last $k - 2$ flips of the above sequence. Given the stack $-I_{n-1}$, we number the pancakes $1, \ldots, j, j + 2, \ldots, n$, and proceed as if a dummy pancake $j + 1$ is initially atop the stack. Then, keeping track of the position of the dummy pancake, we use the final $k - 2$ flips above. In reality, any flip of the top m pancakes which includes the "dummy" actually flips only $m - 1$ pancakes. □

Suppose that we know $T(n) = k$. Then by Theorem 7, we know $T(n + 1) \leqslant k + 2$. Now Theorem 9 tells us that a computer search for a sequence of $k + 1$ flips that sorts $-I_{n+1}$ need not consider any sequence containing a useless pair, since the existence of such a sequence would imply $T(n) \leqslant k - 1$, a contradiction.

6. Improved bounds for sorting $-I_n$

We begin with a lower bound for sorting $-I_n$. Note that since $g(n) \geqslant T(n)$, any lower bound on $T(n)$ also holds for $g(n)$ regardless of whether or not the conjecture is true.

Theorem 10. $T(n), g(n) \geqslant 3n/2$.

Proof. The proof was essentially given in [3], wherein a lower bound of $3n/2 - 1$ is derived on sorting:

$$\begin{bmatrix} n \\ n-1 \\ \vdots \\ 1 \end{bmatrix}$$

Applying Corollary 6.1, we immediately find that $T(n) \geqslant (3n/2 - 1) + 1 = 3n/2$. Since $g(n) \geqslant T(n)$, the lower bound also holds for $g(n)$. □

To improve the upper bound on $T(n)$, we now present a near-optimal recursive algorithm for sorting $-I_n$. The algorithm works by repeated application of sorting sequences for small n.

Recursive algorithm for sorting $-I_n$. Let $t(n)$ be the number of flips required by the recursive algorithm to sort $-I_n$. As the basis for the recursion, suppose that for some m, we have an algorithm (not necessarily optimal) for sorting $-I_m$ in $t(m)$ flips. We can then apply this algorithm recursively to sort $-I_n$ in general by treating the bottom $n - m + 1$ pancakes, numbered m through n, as a single pancake. Without loss of generality, the first flip is taken to be of size n, so the entire stack is turned over. We now remember momentarily that pancakes m through n are distinct, and make use of Corollary 6.1 to recursively sort them in $t(n - m + 1) - 1$ flips. When this is completed, we once again treat them as a single pancake and execute the remaining $t(m) - 1$ flips to complete the sort of the entire stack. The procedure is illustrated below:

$$\begin{bmatrix} -1 \\ \vdots \\ -(m-1) \\ -m \\ \vdots \\ -n \end{bmatrix} \vdash \begin{bmatrix} n \\ \vdots \\ m \\ m-1 \\ \vdots \\ 1 \end{bmatrix} \overset{t(n-m+1)-1}{\vdash} \begin{bmatrix} m \\ \vdots \\ n \\ m-1 \\ \vdots \\ 1 \end{bmatrix} \overset{t(m)-1}{\vdash} \begin{bmatrix} 1 \\ \vdots \\ m-1 \\ m \\ \vdots \\ n \end{bmatrix}$$

We obtain the recurrence

$$\begin{aligned} t(n) &= t(m) - 1 + t(n - m + 1) \\ &= t(m) - 1 + t(n - (m - 1)), \end{aligned}$$

which yields

$$t(n) = \left(\frac{t(m) - 1}{m - 1}\right) \cdot n.$$

At most a constant number of flips will be required to sort the last few pancakes when the recursion bottoms out.

A computer search has located the following sequence of 24 flips to sort $-I_{15}$: (15, 10, 4, 6, 14, 6, 4, 10, 15, 10, 4, 6, 14, 6, 4, 10, 15, 10, 4, 6, 14, 6, 4, 10). The zealous reader may wish to check this by hand. (The sequence consists of three repetitions of an 8 flip subsequence containing a palindrome of length 7; we will not make use of this interesting fact, however.) This gives us a base case for the recursive algorithm with $m = 15$ and $t(m) = 24$, so we obtain

$$t(n) \leqslant \frac{23n}{14} + c$$

for constant c. Since the lower bound on sorting $-I_n$ is $1.5n$, we have shown that $1.5n \leqslant T(n) \leqslant 23n/14 + c \approx 1.6429n + c$. It may be possible to make the upper bound tighter by using the results of more ambitious computer searches as the basis of the recursion.[1]

The $23n/14$ algorithm above is somewhat surprising, since the best known upper bound on sorting *unburnt* pancakes, disregarding additive constants, is $5n/3 > 23n/14$.

---

[1] Heydari and Sudborough [4] have recently located a 48 flip sequence for sorting $-I_{31}$. This improves the bound on the algorithm to $47n/30 + c \approx 1.5667n + c$. The flip sequence is $(31, z, 30, z^R)$, where $z = (18, 12, 24, 20, 26, 22, 6, 12, 14, 18, 24, 30, 14, 12, 18, 31, 26, 12, 22, 16, 12, 10, 6)$, and z^R is the reversal of z. Note that this also strengthens Theorem 11, so that $f(n), g(n) \leqslant 47n/30 + c$ under the conjecture.

Table 1
$f(n)$, $g(n)$, and $T(n)$

n	$f(n)$	$g(n)$	$T(n)$
1	0	1	1
2	1	4	4
3	3	6	6
4	4	8	8
5	5	10	10
6	7	12	12
7	8	14	14
8	9	15	15
9	10	17	17
10	11	18	18
11	13	?	19
12	14	?	21
13	?	?	22
14	?	?	23
15	?	?	24
16	?	?	26
17	?	?	28
18	?	?	29
⋮	⋮	⋮	⋮
24	$\geqslant 27$	?	?

Since any algorithm for sorting burnt pancakes also serves to sort unburnt pancakes, the upper bound on sorting unburnt pancakes is also improved under the conjecture to $23n/14 + c$.

Theorem 11. *Under the conjecture that* $-I_n$ *is the worst case among all stacks of* n *burnt pancakes,* $f(n), g(n) \leqslant 23n/14 + c$.

Note that even if the conjecture is true, the above bound is non-constructive, in the sense that our algorithm does not tell us how to sort any configuration other than $-I_n$.

We conclude with Table 1 containing the exact values of $f(n)$, $g(n)$, and $T(n)$ for some small instances of n, most of which were obtained by computer search. The values of $f(n)$ for $n \leqslant 9$ appeared previously in [3]. The entry for $f(12)$ is from [4], and $T(9)$ and $T(10)$ were obtained in Corollaries 8.1 and 8.2. The lower bound of 27 on $f(24)$ is demonstrated by the stack (topmost pancake on left): $[1,6,3,8,5,2,7,4,9,14,11,16,13,10,15,12,17,22,19,24,21,18,23,20]$, which requires 27 flips to sort.

Acknowledgements

We would like to thank Mohammad Heydari and Hal Sudborough for helpful discussions and for improving on our results by locating a 48 flip sequence for sorting

$-I_{31}$. Thanks also to Mike Reid for his comments, and to the anonymous referees for their highly useful suggestions.

References

[1] S. Akers and B. Krishnamurthy, A group theoretic model for symmetric interconnection networks, in: Proceedings of the 1986 International Conference on Parallel Processing (IEEE, New York, 1986) 216–223.
[2] Amer. Math. Monthly 82 (1975) 1010.
[3] W. Gates and C. Papadimitriou, Bounds for sorting by prefix reversal, Discrete Math. 27 (1979) 47–57.
[4] M. Heydari and H. Sudborough, Private communication.
[5] Z. Miller, D. Pritikin and H. Sudborough, Bounded dilation maps of hypercubes into cayley graphs on the symmetric group (unpublished manuscript, 1992).
[6] K. Qiu, H. Meijer and S. Akl, Parallel routing and sorting on the pancake network, in: Proceedings of the Third International Conference on Computing and Information, Lecture Notes in Computer Science 497 (Springer, Berlin, 1991) 360–371.