Journal of the Nigerian Association of Mathematical Physics
Volume 10 (November 2006), 121 -126
© J of NAMP

Three algorithms for Egyptian fractions

Orobosa Izevbizua and Joseph Osemwenkhae
Department of Mathematics, University of Benin, Benin City

Abstract

The ancient Egyptians used a number system based on unit fractions, i.e. fractions with one in the numerator. This idea let them represent any fraction a/b as the sum of unit fractions e.g $\frac{2}{7}=\frac{1}{4}+\frac{1}{28}$ Further, the same fraction could not be used twice (so $\frac{2}{7}=\frac{1}{7}+\frac{1}{7}$ is not allowed). In this work we examine a number of algorithms for generating Egyptian fractions in more detail, implement them and analyze their performance.

Keywords: Unit fractions, Splitting Algorithms, Paring Algorithm, Distinct divisors, Length of Egyptian fraction, Lexicographic

1.0 Introduction

1.1 Definition

An Egyptian fraction is the sum of positive (usually) distinct unit fractions i.e. expression of the sum of unit fractions like $1/a + 1/b + 1/c + ...$ where the denominators a, b, c, … are increasing. It's interesting to know that every fraction p/q can be represented as a sum of distinct unit fractions [1] and each fraction can be represented in an infinite number of ways. Now consider

$$1 = 1/2 + 1/3 + 1/6 \qquad (*)$$

So if,

$$\frac{3}{4} = \frac{1}{2} + \frac{1}{4} \qquad (1.1)$$

By dividing $*$ by 4, we have that (1.1) becomes

$$\frac{3}{4} = \frac{1}{2} + \frac{1}{8} + \frac{1}{12} + \frac{1}{24} \qquad (1.2)$$

Also by dividing $*$ by 24, (1.2) becomes

$$\frac{3}{4} = \frac{1}{2} + \frac{1}{8} + \frac{1}{12} + \frac{1}{48} + \frac{1}{72} + \frac{1}{144} \qquad (1.3)$$

We can repeat the process by expanding the last term and so on. This shows that once we have found one way of writing p/q as Egyptian fraction, we can derive as many other representations as we wish.

Any number p/q has infinitely many Egyptian fraction representations, although there are only finitely many having a given number of terms [2]. It is not known how the Egyptians found their representations, but today many algorithms are known for this problem, each behaving differently in terms of the number of unit fractions produced, the size of the denominators of the fractions and the time taken to find the representations. Our aim in this work is to examine some algorithms, implement them, analyze their performance and improve on their performance.

2.0 The Splitting Algorithm

This algorithm is based on conflict resolution methods. The algorithm which is based on the repeated use of the equality

$$\frac{1}{x} = \frac{1}{x+1} + \frac{1}{x(x+1)} \tag{2.1}$$

employs the following simple idea: from a fraction p/q we can form a representation in unit fractions by making p copies of $\frac{1}{q}$. This is not an Egyptian fraction since the unit fractions are not distinct. However, we can now search for conflicting pairs (two copies of the same fraction) and resolve the conflict by replacing the pair with some other fractions adding to the same value.

Next we outline the steps involved in the algorithm:

Step 1:- Given, rational $p/q < 1$ in lowest terms

Step 2:- Write p/q as the sum of p unit fractions $1/q$

Step 3:- If there are duplicated fractions $\frac{1}{n}$ in the expansion (for any integer n), keep one of them, but remove the other duplicated $\left(\frac{1}{n}\right)$'s by applying the splitting relation (1.1) to them.

Step 4:- Repeat step 3 until an expansion is reached which has no denominator duplicated.

Example 2.1

Express $\frac{3}{7}$ as an Egyptian fraction.

$$\frac{3}{7} = \frac{1}{7} + \frac{1}{7} + \frac{1}{7}$$

applying (4) to two $\frac{1}{7}$ leaving one, we have

$$\frac{3}{7} = \frac{1}{7} + \frac{1}{8} + \frac{1}{56} + \frac{1}{8} + \frac{1}{56}$$

Again apply (4) to one of $\frac{1}{8}$ & $\frac{1}{56}$, we have

$$\frac{3}{7} = \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{56} + \frac{1}{57} + \frac{1}{72} + \frac{1}{3192} \tag{2.2}$$

The splitting algorithm for Egyptian fractions was first described in 1964 by Stewart [2]. Campbell [3] proves that the splitting algorithm always work for any rational number p/q. The difficulty he had was proving whether or not the algorithm will eventually terminate at some point. In 1991 Laurent Beeckmans [4] proved that the so called splitting algorithm for Egyptian fraction terminates. As part of this work we also provide a proof to back up Beeckmans proof. We note that while it's true that the computation involved in the splitting algorithm can be complex, the fact that it terminates at some point gives it credibility.

As a way of extending the splitting algorithm, we introduce the **pairing algorithm** [8]. Like the splitting algorithm, it uses the conflict resolution idea. But this time, whenever we have a conflicting pair i.e two copies of some fraction $\frac{1}{q}$, we replace them either by a single fraction $\frac{2}{q}$, if q is even or by

$\frac{2}{q+1} + \frac{2}{q(q+1)}$ if q is odd. Note that in either cases, the fractions simplify to have unit numerators and the order in which this is done does not mater. Also the process may combine pairs of fractions to form integers, as we see with sufficiently many copies of $\frac{1}{q}$.

Example 2.2

Express $\frac{3}{7}$ as Egyptian fraction.

$$\frac{3}{7} = \frac{1}{7} + \frac{1}{7} + \frac{1}{7}$$

since q = 7 is odd, we have

$$\frac{3}{7} = \frac{1}{7} + \frac{2}{7+1} + \frac{2}{7(7+1)}$$

$$= \frac{1}{7} + \frac{1}{4} + \frac{1}{28}$$

From example 2, it's obvious that the pairing algorithm produces less number of unit fractions when compared to the splitting algorithm since each replacement of $\frac{1}{q} + \frac{1}{q}$ by $\frac{2}{q}$ reduces the number of terms, initially p, by one, which can happen at most p times. Each other replacement leaves the number of terms the same but reduces the list of terms in lexicographic order; one can only perform such reductions a finite number of times. Therefore the algorithm eventually terminates with a representation having at most p terms.

3-0 Fibonacci–Sylvester algorithm

A much more useful algorithm is the Fibonacci – Sylvester algorithm. It was first discovered by Fibonacci in 1202 [5] and later by Sylvester [6]. The algorithm is a straight forward greedy algorithm, at each step, we simply take the largest unit fraction less than whatever is left. Fibonacci used it, but did not prove that it worked. It was Sylvester who proved its correctness. Below we try to highlight the steps involved in the Fibonacci – Sylvester algorithm

Step 1: Given rational $p/q < 1$ in lowest term

Step 2: Assign $p' = p \text{ and } q' = q$

Step 3: If $p' = 1$ let p'/q' be part of the expansion and we are done; otherwise use the division algorithm to obtain $q' = Sp' + r$ where $r < p$

Step 4: Note that $\frac{p'}{q'} = \frac{1}{s+1} + \frac{p'-r}{q'(s+1)}$.. So let $\frac{1}{s+1}$ be part of the expansion

Step 5: Let $p'' = p' - r$ and $q'' = q'(s+1)$

Step 6: Reduce $\frac{p''}{q''}$ to lowest terms and go back to step 3.

Example 2.3: Express $\frac{3}{7}$ as Egyptian fraction.

Solution

Given $\frac{3}{7} < 1$

3 =3, 7 = 7

$7 = 2.3 + 1 \equiv q' = sp' + r$

$$\therefore \frac{3}{7} = \frac{1}{2+1} + \frac{3-1}{7(2+1)} = \frac{1}{3} + \frac{2}{21}$$

Now $21 = 10.2 + 1$

$$\therefore \frac{2}{21} = \frac{1}{10+1} + \frac{2-1}{21(11)} = \frac{1}{11} + \frac{1}{231}$$, therefore $\frac{3}{7} = \frac{1}{3} + \frac{1}{11} + \frac{1}{231}$.

Next we produce a proof that the Fibonacci–Sylvester algorithm produces p or fewer terms for any rational $\frac{p}{q},\ q \neq 0$.

Theorem 2.1

The Fibonacci – Sylvester algorithm produces an expansion with p or fewer terms for any rational $\frac{p}{q},\ q \neq 0$.

Proof

The algorithm produces $$\frac{p'}{q'} = \frac{1}{s+1} + \frac{p'-r}{q'(s+1)}$$

Intuitively, trivially the algorithm produces at most *p* terms because the numerators always get smaller.

Now since $\frac{p'}{q'}$ is in lowest terms we know that $r > 0$. At step 5 we have $p'' = p' - r$, so the new $p'' \le$ odd $p' - 1$.

At step 3, we stop if $p' = 1$, so there can be at most *p* terms. Thus, the worst case is where r = 1 each time, and the resulting fraction is always in lowest terms. Then the expansion clearly produces p terms. The major problem, associated with the Fibonacci – Sylvester algorithm is that the denominators can grow quite huge making computation difficult.

3.0 Binary algorithm

First we note that if $N = 2^n$, then any m < N can be written as the sum of distinct divisors of N. We can write the numbers in binary notation. Infact, m can be written as the sum of n or less divisors, since 2^n has exactly n divisors i.e $2^0, 2^1, 2^2, \ldots, 2^{n-1}$. Next, we list the steps involved in the Binary algorithm for Egyptian fractions

Step 1: Given rational $p/q < 1$ in lowest term.

Step 2: Find $N_{k-1} < q \le N_k$ where $N_k = 2^k$.

Step 3: If $q = N_k$ then write out p as the sum of k or less divisors of N_k. $p = \sum_{i=1}^{1} d_i$ and get the expansion

$$\frac{p}{q} = \sum_{i=1}^{j} \frac{d_i}{N_k} = \sum_{i=1}^{j} \frac{1}{N_{k/d_i}}$$

Otherwise go to step 4

Step 4: By dividing PN_k by q we find s and r satisfying $qs + r = PN_k$, where $0 < r < N_k$ such that

$$\frac{p}{q} = \frac{PN_k}{qN_k} = \frac{qs+r}{qN_k} = \frac{S}{N_k} + \frac{r}{qN_k}$$

Step 5: Write $S = \sum d_i$ where d_i are distinct divisors of N_k and write $r = \sum d_i'$ *where* d_i are distinct divisors of N_k.

Step 6: Thus we get the expansion $$\sum \frac{1}{N_k/d_i} + \sum \frac{1}{qN_k/d_i}$$

Example 4

Express $\frac{5}{21}$ as Egyptian fraction.

Solution

Given $\frac{5}{21}$. Observe that $N_k = 2^k = 2^5 = 32$,

$N_{k-1} = 2^{k-1} = 2^4 = 16 \Rightarrow N_{k-1} < q \le N_k = 16 < 21 < 32$. Now $q = 21 \ne N_k = 32$, so we go to step 4

$$\frac{5}{21} = \frac{5(32)}{21(32)}$$

$$\frac{7(21)+13}{21(32)}$$

$$\frac{7}{32} + \frac{13}{21(32)}$$

Note that $\begin{matrix} 7 = 1+2+4 \\ 13 = 1+4+8 \end{matrix}$ {by step 5}. By step 6 we have,

$$\frac{5}{21} = \frac{1}{32/4} + \frac{1}{32/2} + \frac{1}{32/1} + \frac{1}{21.32/8} + \frac{1}{21.32/4} + \frac{1}{21.32/1} + 1/8 + 1/6 + 1/32 + 1/84 + 1/168 + 1/672$$

Also we observe that the division by q ensures that no overlap occurs between the fractions from the two parts of the representation in step 6. Next, we produce a proof that the Binary algorithm produces a finite number of terms.

Theorem 3.1

The Binary algorithm for Egyptian fractions is guaranteed to produce an expansion with $D(n) < n^2$ and $L(n) = 0\,(\log n)$, *where* $D(n)$ and $L(n)$ *represent denominators and length respectively. Simply put the Binary algorithm terminates.*

Proof:

In step 3, note that $p < q < N_k$ so $PN_k < qN_k$, $qs + r = PN_k < qN_k$. So $S < N_k$. This we can always find an expansion for both s and r. The resulting denominators of the expansion are distinct because q divides the second set of denominators (corresponding to *r*). Corresponding to s unless q is a power of 2. But if it were, we never would have gotten past step 2. So the algorithm works. In the case where $q = N_k$, the expansion clearly has at most k terms. In the case where $q < N_k$, the expansion has at most 2k terms. Since $k = \log_2 N_k$, it follows that there are at most $2\log q$ terms in the expansion. Thus, $L(n) = 0\,(\log n)$. In the case where $q < N_k$, the largest denominator can be qN_k, so the largest denominator must be at most $q(q-1)$. Thus $D(n) = 0(n^2)$.

4.0 Conclusion

Nowadays, we usually write non-integer numbers either as fraction $(2/7)$ or decimals (0.285714). But the ancient Egyptians used a number system based on unit fractions. Many algorithms now exist for generating Egyptian fractions. We have in this work examined three algorithms, which include the splitting, Fibonacci-Sylvester and Binary algorithms. These algorithms all have a common problem that is, producing infinite number of terms which do not terminate. In line with our aim of improving these algorithms, we have produced theorems to bound the number of terms produced by these algorithms. Also, we have refined the number of terms produced by the splitting algorithm by introducing the pairing algorithm.

References

[1] Hoffman K. (1998). On Egyptian fractions, Journal of number theory, Vol 43, pp.154 – pp.160.

[2] Stewart I. (1992). The riddle of the vanishing camel, Scientific American, Vol. 80 pp.122 – pp.124.

[3] Campbell K. (1997). Egyptian Mathematics, Journal of number of theory, Vol. 45, pp.134 – pp.136.

[4] Beeckmans L. (1993). The Splitting algorithm for Egyptian fractions, Journal of number theory, Vol. 43 pp.173 – pp.185.

[5] Dunce T. (1996). Egyptian fractions. Bull. London Math Soc, Vol. 66 pp.21 – pp 36.

[6] Sylvester K. (1999). Egyptian fractions, Journal of number theory Vol. 35, pp.120 – pp.135.

[7] Bleicher M. N. (1972). A new algorithm for the expansion of continued fractions, journal of number theory Vol. 4, pp.342 – pp.382.

[8] David Eppstein (1995). Algorithms for Egyptian Fractions. Mathematica in Education a