Farey Sequence & Sums

By *Jinxian Lu, Sebastian Dabrowski, Vivienne Zhu*

May 5th, 2016

Definition of Farey Series

A Farey series is an increasing sequence of order *N*. Each fraction *a/b* (with 0 < a <= b and 1 < b <= N) in the sequence must be in its simplest form and therefore a and b are relatively prime in relation to each other.

Order	Farey Series	Count
1	$\frac{0}{1}$ $\frac{1}{1}$	2
2	$\frac{0}{1}$ $\frac{1}{2}$ $\frac{1}{1}$	3
3	$\frac{0}{1}$ $\frac{1}{3}$ $\frac{1}{2}$ $\frac{2}{3}$ $\frac{1}{1}$	5
4	$\frac{0}{1}$ $\frac{1}{4}$ $\frac{1}{3}$ $\frac{1}{2}$ $\frac{2}{3}$ $\frac{3}{4}$ $\frac{1}{1}$	7
5	$\frac{0}{1}$ $\frac{1}{5}$ $\frac{1}{4}$ $\frac{1}{3}$ $\frac{2}{5}$ $\frac{1}{2}$ $\frac{3}{5}$ $\frac{2}{3}$ $\frac{3}{4}$ $\frac{4}{5}$ $\frac{1}{1}$	11
6	$\frac{0}{1}$ $\frac{1}{6}$ $\frac{1}{5}$ $\frac{1}{4}$ $\frac{1}{3}$ $\frac{2}{5}$ $\frac{1}{2}$ $\frac{3}{5}$ $\frac{2}{3}$ $\frac{3}{4}$ $\frac{4}{5}$ $\frac{5}{6}$ $\frac{1}{1}$	13

Theorem

Order	Farey Series	Count
1	$\frac{0}{1}\ \frac{1}{1}$	2
2	$\frac{0}{1}\ \frac{1}{2}\ \frac{1}{1}$	3
3	$\frac{0}{1}\ \frac{1}{3}\ \frac{1}{2}\ \frac{2}{3}\ \frac{1}{1}$	5
4	$\frac{0}{1}\ \frac{1}{4}\ \frac{1}{3}\ \frac{1}{2}\ \frac{2}{3}\ \frac{3}{4}\ \frac{1}{1}$	7
5	$\frac{0}{1}\ \frac{1}{5}\ \frac{1}{4}\ \frac{1}{3}\ \frac{2}{5}\ \frac{1}{2}\ \frac{3}{5}\ \frac{2}{3}\ \frac{3}{4}\ \frac{4}{5}\ \frac{1}{1}$	11
6	$\frac{0}{1}\ \frac{1}{6}\ \frac{1}{5}\ \frac{1}{4}\ \frac{1}{3}\ \frac{2}{5}\ \frac{1}{2}\ \frac{3}{5}\ \frac{2}{3}\ \frac{3}{4}\ \frac{4}{5}\ \frac{5}{6}\ \frac{1}{1}$	13

Theorem 1. Definition of Neighbors

In any Farey series, if *a/b* and *a'/b'* are two adjacent terms in which *a/b* < *a'/b'*, then *a'b - ab'* = *1*. For instance, in the Farey sequence of order 6, 1/3 and 2/5 are two adjacent terms with 1/3 < 2/5. Then (2 * 3) - (1* 5) = 6 - 5 = 1.

Theorem 2. Finding the Complementary Fraction

A Farey series contains complementary fractions. If a/b is in a Farey series of order N, then a/b + (b-a)/b = 1. For instance, in the Farey sequence of order 3, 0/1 and 1/1, 1/3 and 2/3 are complementary fractions respectively.

Theorem 3. Finding the Mediant

In a Farey series, between fractions a/b and a'/b' lies the fraction *(a+a') / (b+b')* that is formed by summing the numerators and denominators:

$$a/b < (a+a')\,/\,(b+b') < a'/b'$$

For instance, when *N* = 3, if *a/b* = 1/3 and *a'/b'* = 2/3, then we can generate a new fraction between 1/3 and 2/3 : (1+2) / (3+3) = 3/6 = 1/2 which is larger than 1/3 but smaller than 2/3.

Farey Sum

If the denominators of the Farey series of order *N* are:

d[1], d[2], ... , d[K]

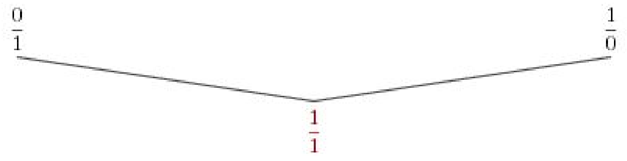
then the Farey sum of order *N* is the sum of *d[i] / d[i+1]* from *i* = 1 to *K* - 1. Half of such terms are the reciprocal of the other half of the terms. For instance, when *N* = 3, the terms *d[i] / d[i+1]* are 1/3, 3/2, 2/3, and 3/1. The reciprocal of 1/3 is 3/1 and the reciprocal of 3/2 is 2/3.

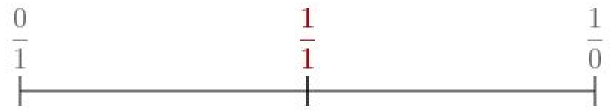
Order	Farey Sum
1	1/1
2	1/2 + 2/1 =5/2
3	1/3 + 3/2 + 2/3 + 3/1 = 11/2
4	1/4 + 4/3 +3/2 + 2/3 + 3/4 + 4/1 = 17/2
5	1/5 + 5/4 + 4/3 + 3/5 + 5/2 + 2/5 + 5/3 + 3/4 + 4/5 + 5/1 = 29/2
6	1/6 + 6/5 + 5/4 + 4/3 + 3/5 + 5/2 + 2/5 + 5/3 + 3/4 + 4/5 + 5/6 + 6/1 = 35/2

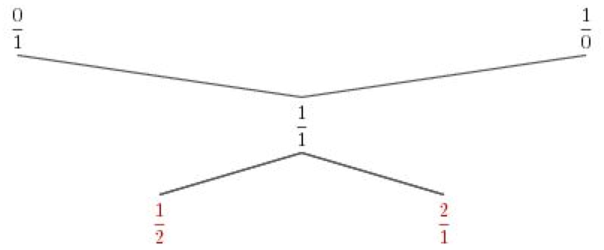
Stern–Brocot Tree

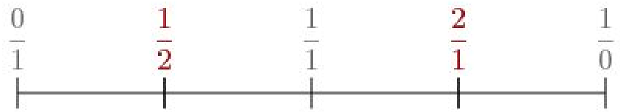
The Farey series has a close relationship with the Stern-Brocot tree since they both have the same property that is discussed in **Theorem 3** (define the mediant of the two rational numbers).

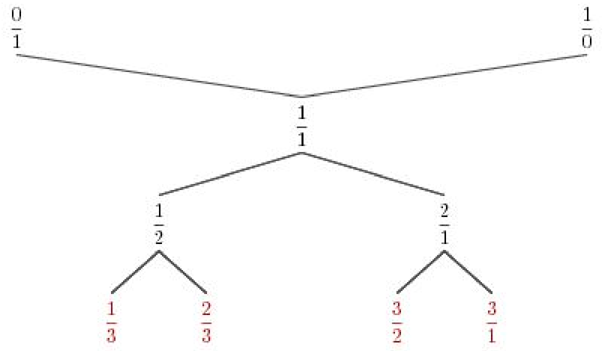
The construction of the Stern-Brocot tree begins with the two fractions 0/1 and 1/0. To continue the construction, we insert the mediant of any two consecutive rational numbers in the tree.

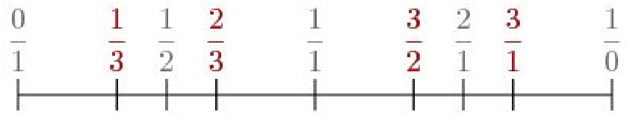




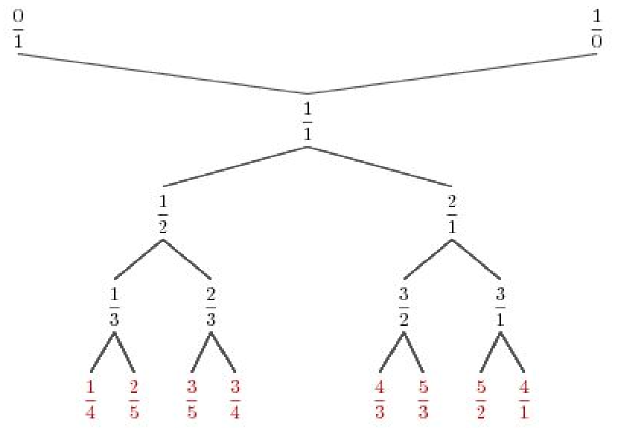


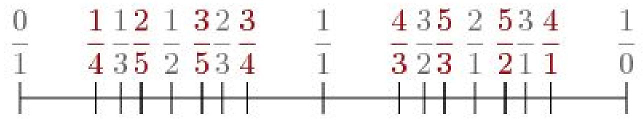


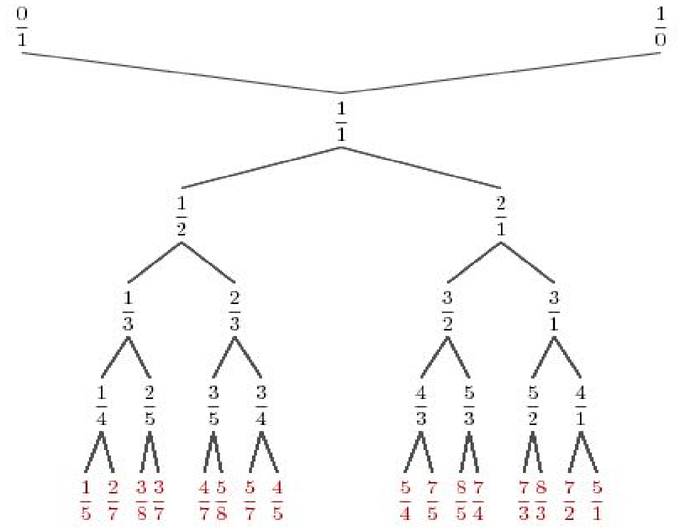




Continue...

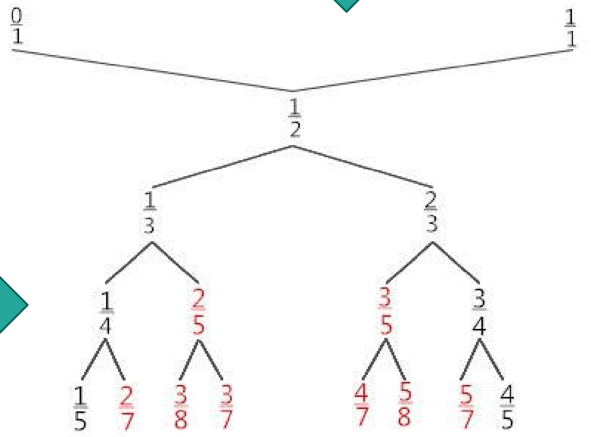






As shown in the figure, the fractions in the Farey series of order *N* from 1 to 5 are included in the Stern-Brocot tree. Since the Farey series only generates fractions from 0 to 1, instead of beginning with 0/1 and 1/0, we begin with 0/1 and 1/1 to construct the tree. So the right half of the Stern-Brocot tree will be cancelled.

The denominators in the Farey series of order *N* do not exceed *N*, so the fractions in red will not be counted in the Farey series



The First Algorithm

In the first program our group wrote, we tried to stay as close to the original instructions as possible. This meant building a list of the Farey Sequence and then using it to create and add up the Farey sum. To build the Farey Sequence, we used strong induction and a version of the third theorem of the Farey Sequence.

Observe the Farey Sequence of order 5 and 6:

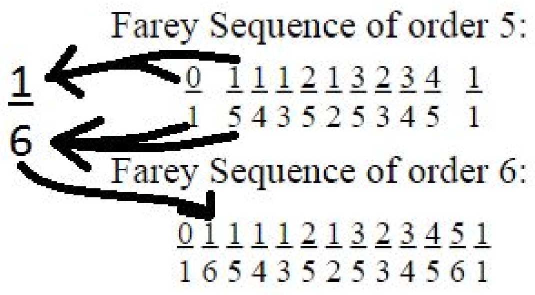
Farey Sequence of order 5:

$$\frac{0}{1}\ \ \frac{1}{5}\,\frac{1}{4}\,\frac{1}{3}\,\frac{2}{5}\,\frac{1}{2}\,\frac{3}{5}\,\frac{2}{3}\,\frac{3}{4}\,\frac{4}{5}\ \ \frac{1}{1}$$

Farey Sequence of order 6:

$$\frac{0}{1}\,\frac{1}{6}\,\frac{1}{5}\,\frac{1}{4}\,\frac{1}{3}\,\frac{2}{5}\,\frac{1}{2}\,\frac{3}{5}\,\frac{2}{3}\,\frac{3}{4}\,\frac{4}{5}\,\frac{5}{6}\,\frac{1}{1}$$

The difference between these two sequences is that order 6 has fractions in it with a denominator of 6. In order to find these fractions all we have to do is go through the sequence of order 5 step by step and find denominators which add up to 6. When we find two that do, we insert a new fraction in between with a denominator of 6 and a numerator equal to the sum of those fractions numerators.



The First Algorithm(continued)

Building the Farey Sum

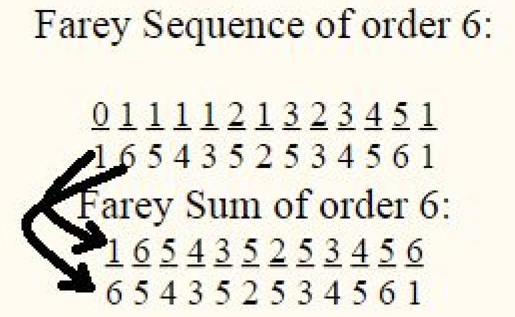
Once the Farey Sequence was constructed, we used the denominators of the fractions in the sequence to find the fractions of the Farey Sum. We do this by starting at the beginning of the sequence and taking the denominator of the first and second fractions. These become the first fraction in the Farey Sum. The process is repeated until the end of the sequence is reached. Once all the fractions in the Farey Sum are found, we are able to add them up and return the Farey Sum.

Farey Sequence of order 6:

$$\frac{0}{1}\ \frac{1}{6}\ \frac{1}{5}\ \frac{1}{4}\ \frac{1}{3}\ \frac{2}{5}\ \frac{1}{2}\ \frac{3}{5}\ \frac{2}{3}\ \frac{3}{4}\ \frac{4}{5}\ \frac{5}{6}\ \frac{1}{1}$$

Farey Sum of order 6:

$$\frac{1}{6}\ \frac{6}{5}\ \frac{5}{4}\ \frac{4}{3}\ \frac{3}{5}\ \frac{5}{2}\ \frac{2}{5}\ \frac{5}{3}\ \frac{3}{4}\ \frac{4}{5}\ \frac{5}{6}\ \frac{6}{1}$$



Farey Sequence of order 6:

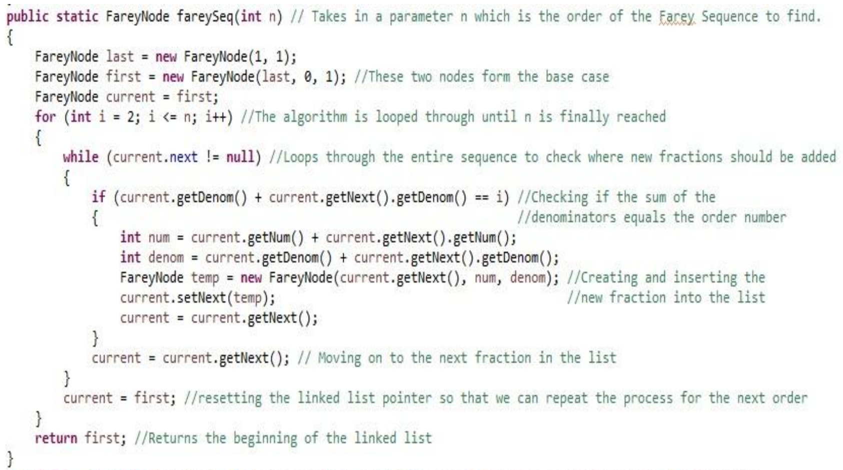
$$\frac{0}{1}\ \frac{1}{6}\ \frac{1}{5}\ \frac{1}{4}\ \frac{1}{3}\ \frac{2}{5}\ \frac{1}{2}\ \frac{3}{5}\ \frac{2}{3}\ \frac{3}{4}\ \frac{4}{5}\ \frac{5}{6}\ \frac{1}{1}$$

Farey Sum of order 6:

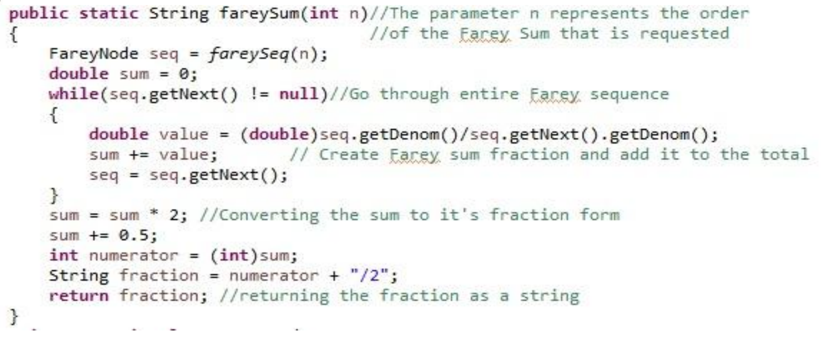
$$\frac{1}{6}\ \frac{6}{5}\ \frac{5}{4}\ \frac{4}{3}\ \frac{3}{5}\ \frac{5}{2}\ \frac{2}{5}\ \frac{5}{3}\ \frac{3}{4}\ \frac{4}{5}\ \frac{5}{6}\ \frac{6}{1}$$

Final answer for order 6: 35/2

The First Algorithm - Coding & Time complexity



```
public static FareyNode fareySeq(int n) // Takes in a parameter n which is the order of the Farey Sequence to find.
{
    FareyNode last = new FareyNode(1, 1);
    FareyNode first = new FareyNode(last, 0, 1); //These two nodes form the base case
    FareyNode current = first;
    for (int i = 2; i <= n; i++) //The algorithm is looped through until n is finally reached
    {
        while (current.next != null) //Loops through the entire sequence to check where new fractions should be added
        {
            if (current.getDenom() + current.getNext().getDenom() == i) //Checking if the sum of the
            {                                                            //denominators equals the order number
                int num = current.getNum() + current.getNext().getNum();
                int denom = current.getDenom() + current.getNext().getDenom();
                FareyNode temp = new FareyNode(current.getNext(), num, denom); //Creating and inserting the
                current.setNext(temp);                                         //new fraction into the list
                current = current.getNext();
            }
            current = current.getNext(); // Moving on to the next fraction in the list
        }
        current = first; //resetting the linked list pointer so that we can repeat the process for the next order
    }
    return first; //Returns the beginning of the linked list
}
```



```
public static String fareySum(int n)//The parameter n represents the order
{                                    //of the Farey Sum that is requested
    FareyNode seq = fareySeq(n);
    double sum = 0;
    while(seq.getNext() != null)//Go through entire Farey sequence
    {
        double value = (double)seq.getDenom()/seq.getNext().getDenom();
        sum += value;       // Create Farey sum fraction and add it to the total
        seq = seq.getNext();
    }
    sum = sum * 2; //Converting the sum to it's fraction form
    sum += 0.5;
    int numerator = (int)sum;
    String fraction = numerator + "/2";
    return fraction; //returning the fraction as a string
}
```

After we figured out the algorithm we were going to try first, coding was relatively quick. We used a linked list to store the Farey sequence because it ended up being more time efficient in our case than an array.

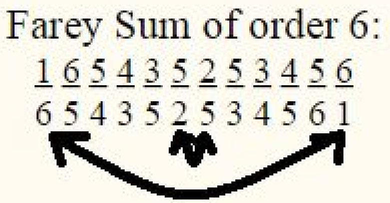
After all the coding was done, we analyzed the time complexity of the algorithm. What we came up with was that our algorithm's time complexity is equal to:

$$\left(\sum_{1}^{n} f(x)\right) + f(n)$$

Here f(x) represents the length of a Farey sequence of order x.

The Second Algorithm

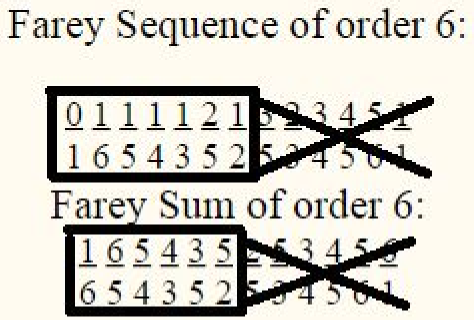
The second algorithm was focused on trying to improve on the time complexity of the first algorithm. To do this we used the fact that the second half of the fractions of the Farey Sum are reciprocals of the first half.



Farey Sum of order 6:

$$\frac{1}{6}\ \frac{6}{5}\ \frac{5}{4}\ \frac{4}{3}\ \frac{3}{5}\ \frac{5}{2}\ \frac{2}{5}\ \frac{5}{3}\ \frac{3}{4}\ \frac{4}{5}\ \frac{5}{6}\ \frac{6}{1}$$

With this property we can effectively stop storing half the Farey Sequence that we were storing before without encountering any problems.



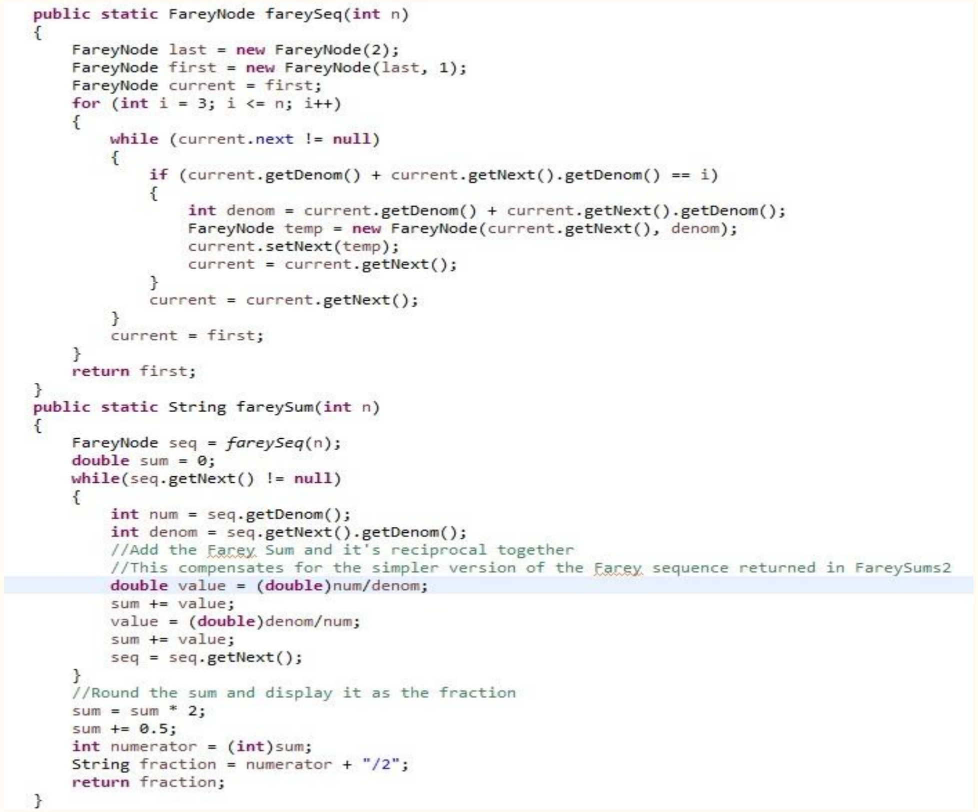
Farey Sequence of order 6:

$$\boxed{\frac{0}{1}\ \frac{1}{6}\ \frac{1}{5}\ \frac{1}{4}\ \frac{1}{3}\ \frac{2}{5}\ \frac{1}{2}}\ \cancel{\frac{3}{5}\ \frac{2}{3}\ \frac{3}{4}\ \frac{4}{5}\ \frac{5}{6}\ \frac{1}{1}}$$

Farey Sum of order 6:

$$\boxed{\frac{1}{6}\ \frac{6}{5}\ \frac{5}{4}\ \frac{4}{3}\ \frac{3}{5}\ \frac{5}{2}}\ \cancel{\frac{2}{5}\ \frac{5}{3}\ \frac{3}{4}\ \frac{4}{5}\ \frac{5}{6}\ \frac{6}{1}}$$

Algorithm 2 - Coding and Time Complexity



```java
public static FareyNode fareySeq(int n)
{
    FareyNode last = new FareyNode(2);
    FareyNode first = new FareyNode(last, 1);
    FareyNode current = first;
    for (int i = 3; i <= n; i++)
    {
        while (current.next != null)
        {
            if (current.getDenom() + current.getNext().getDenom() == i)
            {
                int denom = current.getDenom() + current.getNext().getDenom();
                FareyNode temp = new FareyNode(current.getNext(), denom);
                current.setNext(temp);
                current = current.getNext();
            }
            current = current.getNext();
        }
        current = first;
    }
    return first;
}
public static String fareySum(int n)
{
    FareyNode seq = fareySeq(n);
    double sum = 0;
    while(seq.getNext() != null)
    {
        int num = seq.getDenom();
        int denom = seq.getNext().getDenom();
        //Add the Farey Sum and it's reciprocal together
        //This compensates for the simpler version of the Farey sequence returned in FareySums2
        double value = (double)num/denom;
        sum += value;
        value = (double)denom/num;
        sum += value;
        seq = seq.getNext();
    }
    //Round the sum and display it as the fraction
    sum = sum * 2;
    sum += 0.5;
    int numerator = (int)sum;
    String fraction = numerator + "/2";
    return fraction;
}
```

By halving the length of the Farey Sequence stored, we improve our time complexity significantly.

$$\frac{\left(\sum_{1}^{n} f(x)\right) + f(n)}{2}$$

With this change, the time taken to find a solution is effectively halved, since the loops in the program only have to loop through approximately half of what they did before. However, even with this improvement, the algorithm is still not ideal.

The Third Algorithm

When our group got to making a third algorithm, we knew that we had to find a way other than strong induction if we wanted a time efficient algorithm. However, we were stuck for a while. Eventually we decided to look at the ACM programming contest solution and in there we found what we were missing. The Farey Sum has a property where it increases by 3/2 for every element in the Farey Sequence that is not 0/1 or 1/1.

FareySum = 1 + (3/2 * (FareySeqenceLength - 2))

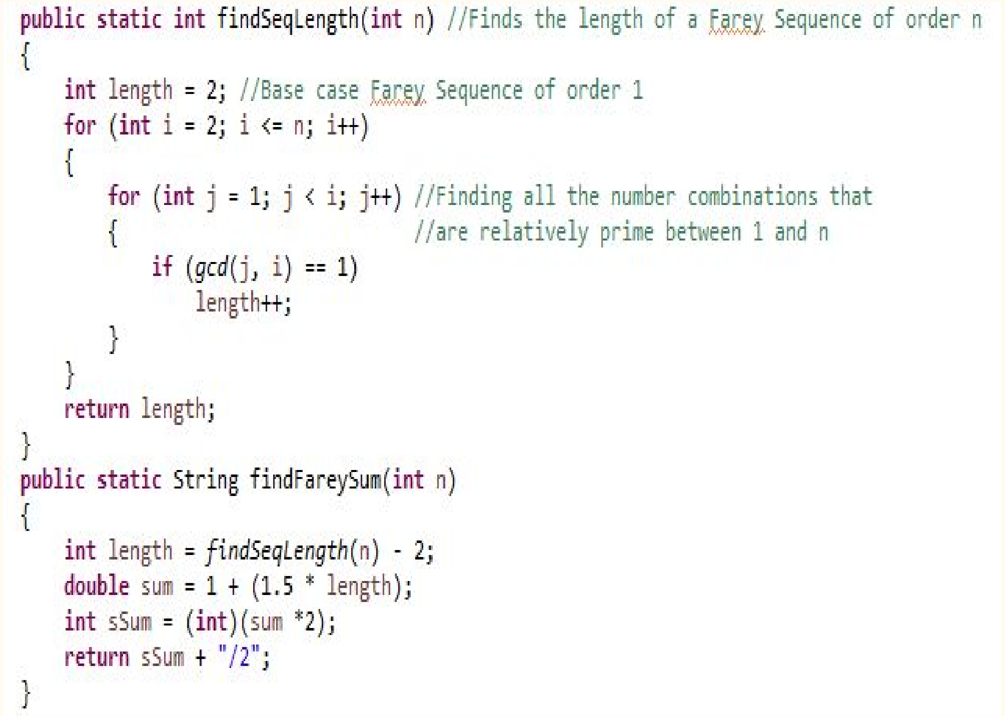
Once we found this equation, the problem became a lot different. We no longer needed the fractions of the Farey sequence to find the Farey Sum. We only needed the sequence's length. So how do we find the length without building it? Well the Farey sequence is basically a set of all the simplified fractions between 0 and 1. Their numerators and denominators are all relatively prime relative to each other.

Farey Sequence of order 6:

$$\frac{0}{1}\ \frac{1}{6}\ \frac{1}{5}\ \frac{1}{4}\ \frac{1}{3}\ \frac{2}{5}\ \frac{1}{2}\ \frac{3}{5}\ \frac{2}{3}\ \frac{3}{4}\ \frac{4}{5}\ \frac{5}{6}\ \frac{1}{1}$$

This means that if we loop through all the possible pairs of numbers from 1 - n, counting the pairs of numbers that have a gcd of 1, we will find the length of the Farey sequence and therefore the Farey Sum.

Algorithm 3 - Coding and Time Complexity



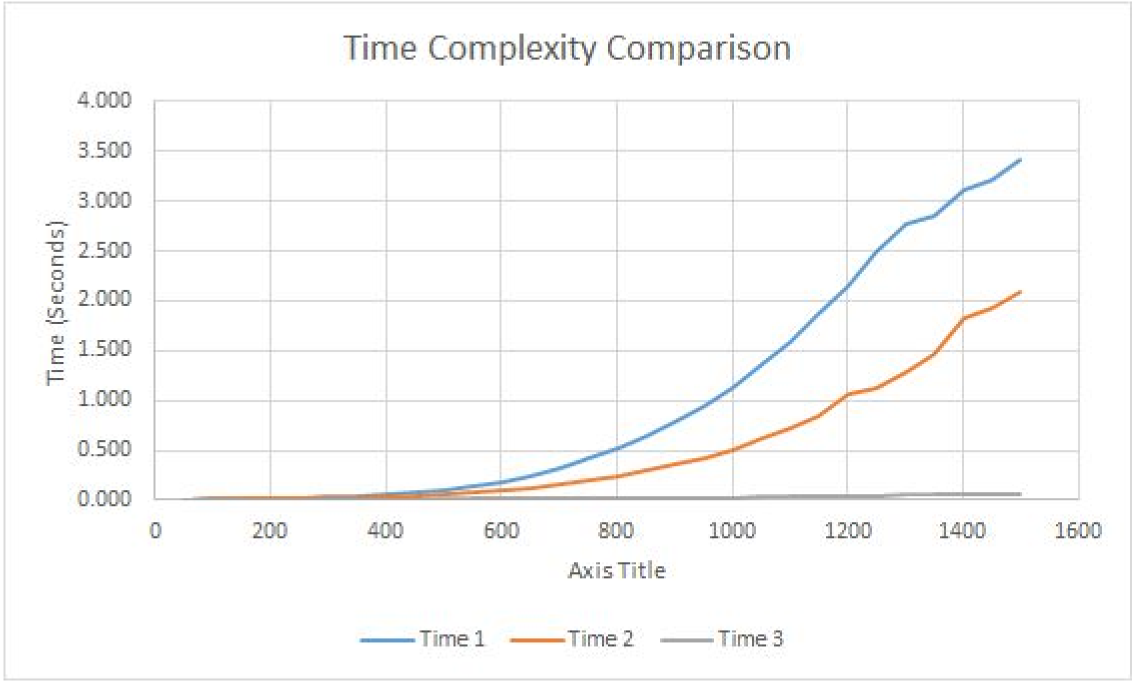
```
public static int findSeqLength(int n) //Finds the length of a Farey Sequence of order n
{
    int length = 2; //Base case Farey Sequence of order 1
    for (int i = 2; i <= n; i++)
    {
        for (int j = 1; j < i; j++) //Finding all the number combinations that
        {                           //are relatively prime between 1 and n
            if (gcd(j, i) == 1)
                length++;
        }
    }
    return length;
}
public static String findFareySum(int n)
{
    int length = findSeqLength(n) - 2;
    double sum = 1 + (1.5 * length);
    int sSum = (int)(sum *2);
    return sSum + "/2";
}
```

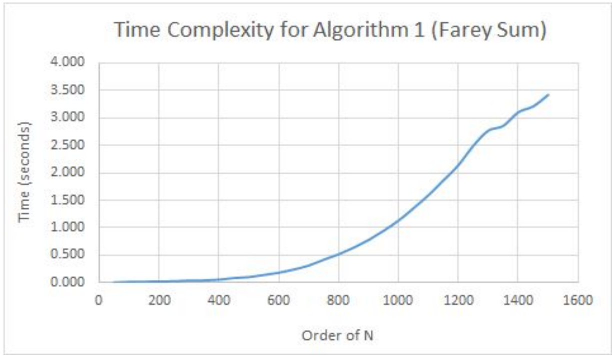
After analyzing the code of our algorithm, the time complexity turns out to be just:

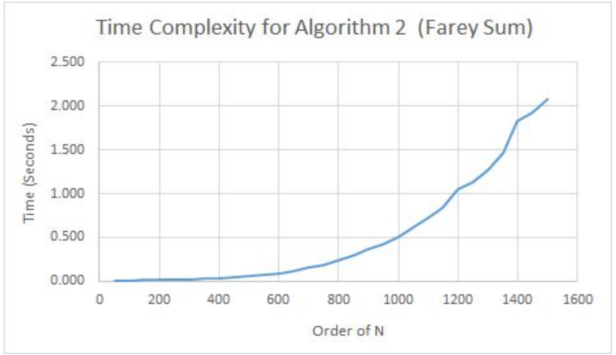
$$\sum_{i=1}^{n} i$$

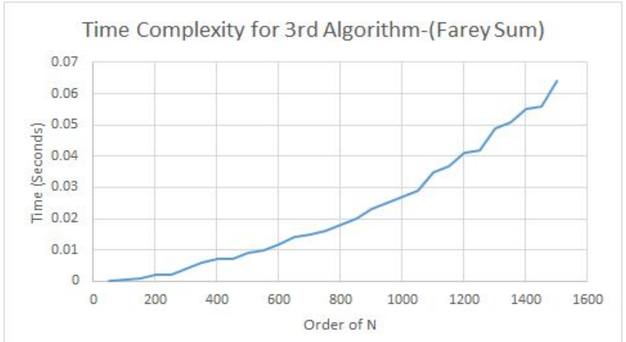
This of course simplifies to (n(n+1))/2. After multiplying out the n, this can be further simplified down to Θ(n^2). This makes it the fastest and most efficient of our 3 algorithms.

Time Complexity Comparison







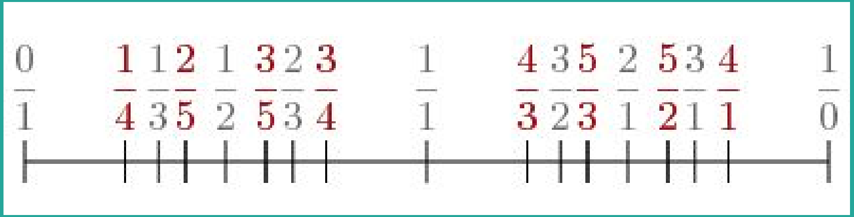


Application of Farey Sequence

Find rational approximation of irrational numbers

Theorem 3

For two positive rational numbers b/d and a/c, if 0 < b/d < a/c, the mediant is a+b/c+d, which means b/d < (a+b) / (c+d) < a/c



Applying the theorem in finding approximation:

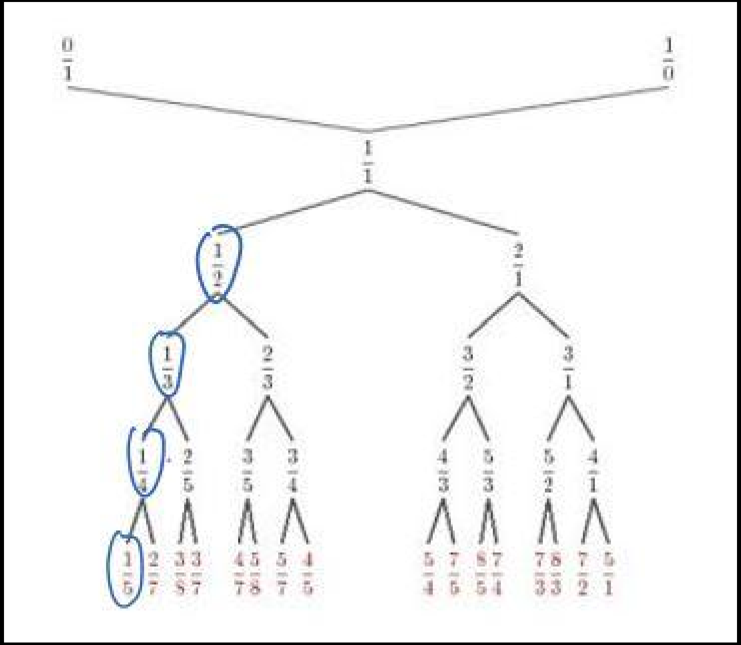
Example

To find out the closet rational approximation of 1/***π*** where denominator is no greater than 100.

Solution:

{0/1, 1/1 } is the first interval and the first Farey sequence to start with and the mediant in this interval is ½. As it is clear that ***π*** >2, therefore, 1/***π*** < ½, which means 1/***π*** lies between the interval of {0/1, ½}. The mediant for {0/1, ½} now becomes (0+1)/(1+2), which is 1/3. Repeating this step until the denominator is approaching 100.

Implementation: Stern-Brocot Tree


$\frac{0}{1}$
$\frac{1}{0}$
$\frac{1}{1}$
$\frac{1}{2}$
$\frac{2}{1}$
$\frac{1}{3}$
$\frac{2}{3}$
$\frac{3}{2}$
$\frac{3}{1}$
$\frac{1}{4}$
$\frac{2}{5}$
$\frac{3}{5}$
$\frac{3}{4}$
$\frac{4}{3}$
$\frac{5}{3}$
$\frac{5}{2}$
$\frac{4}{1}$
$\frac{1}{5}$ $\frac{2}{7}$ $\frac{3}{8}$ $\frac{3}{7}$ $\frac{4}{7}$ $\frac{5}{8}$ $\frac{5}{7}$ $\frac{4}{5}$ $\frac{5}{4}$ $\frac{7}{5}$ $\frac{8}{5}$ $\frac{7}{4}$ $\frac{7}{3}$ $\frac{8}{3}$ $\frac{7}{2}$ $\frac{5}{1}$

The computer will go through the Stern-Brocot tree and narrow down to one fraction number when it meets the requirement. For example, it will go the left side of {0/1, 1/1} to further search for a rational fraction for 1/***π***. In this way, it will take a very short time to find out the answer by incorporating the Stern-Brocot tree. This is similar to the divide and conquer method, which narrows the search range into one half of its original, and by repeating this step, the answer can be found within a second.

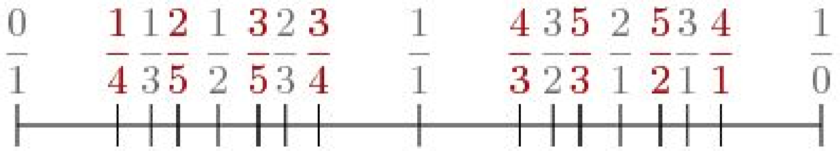
Drawbacks and Time Complexity

- The irrational number is limited to between 0 and 1.
- The best approximation does not guarantee the most appropriate solution.
- The exact value of irrational number has to be known in order to find a better approximation.

1/***π*** = 0.31830988618

The best approximation is 29/91 = 0.3186813186813187

The most accurate approximation is 7/22 = 0.31818181818



$\frac{0}{1}$ $\frac{1}{4}$ $\frac{1}{3}$ $\frac{2}{5}$ $\frac{1}{2}$ $\frac{3}{5}$ $\frac{2}{3}$ $\frac{3}{4}$ $\frac{1}{1}$ $\frac{4}{3}$ $\frac{3}{2}$ $\frac{5}{3}$ $\frac{2}{1}$ $\frac{5}{2}$ $\frac{3}{1}$ $\frac{4}{1}$ $\frac{1}{0}$

- Having the Stern-Brocot tree will tremendously save time, however the prerequisite is that the tree needed to be built up first.
- Time complexity for finding the best approximation

T(n) = O log (n)

Conclusion

Our Research

Our team has explored the historical background of the Farey Sum and studied the theorems behind it. The Farey Sequence is displayed in Stern-Brocot Tree, which allows us to further dig out relationships between fractions.

Three Algorithms

Our team created three algorithms to compute the Farey Sum and continued to improve the algorithms by making them simpler and more efficient with each iteration. As shown above, the algorithms have used various Farey Sum properties and programming methods in order to try and create the most efficient algorithm.

Mathematical Application

In addition to algorithms, our team has also demonstrated how to utilize the properties of Farey Sums to solve practical mathematical problems. The time to find the rational approximation of irrational numbers is almost constant by turning the problem into a Farey Sequence problem.

In all, there may be other better algorithms and applications for the Farey Sequence but what remain unchanged are those common theorems in the Farey Sequence that we have studied through this project.

References

- Guthery, S. B. (2011). *A motif of mathematics*:. Boston: Docent Press.
- Daintith, J., & Nelson, R. D. (1989). *The Penguin dictionary of mathematics*. London: Penguin Books.
- Fractions in the Farey Series and the Stern-Brocot Tree. (n.d.). Retrieved April 17, 2016, from http://www.maths.surrey.ac.uk/hosted-sites/R.Knott/Fractions/fareySB.html#section1.1
- Austin, D. (n.d.). Feature Column from the AMS. Retrieved April 22, 2016, from http://www.ams.org/samplings/feature-column/fcarc-stern-brocot
- Dawson, Jonathan Ainsworth Michael, and John Pianta James Warwick. (n.d.). “The Farey Sequence.” Retrieve May 2, 2016, from http://www.maths.ed.ac.uk/~aar/fareyproject.pdf

T H A N K Y O U