Following is the algorithm to find friends:

1: Create an array of integers of size max limit

2. Initially let i = 1, the default number for summation to the result

3: Start the second loop which runs for all the multiples of ‘i’, keep adding them to the result

4: For every number till max limit, find the friends pair, whose sum is less than the number and print it: in this way printing duplicate pairs is avoided

Code Snippet from Friends.java:

**for**(**int** i=1;i<max;i++){

**for**(**int** j=i\*2;j<max;j=j+i){

sumFactor[j]+=i;

}

**if**((sumFactor[i])<i && sumFactor[sumFactor[i]]==i ){

System.***out***.println(counter++ + " : "+(sumFactor[i]) +" and "+i);

}

}

Time Complexity:

The upper loop is getting executed from 1->n times, while the inner loop goes as follows:

For i=1: j=2->n with increment of 1 ~ n

For i=2: j=4->n with increment of 2 ~ n/2

For i=3: j=6->n with increment of 3 ~ n/3 and so on...

So overall code complexity will be n(n+n/2+n/3+n/4+...+1) ~ n(2n-1)