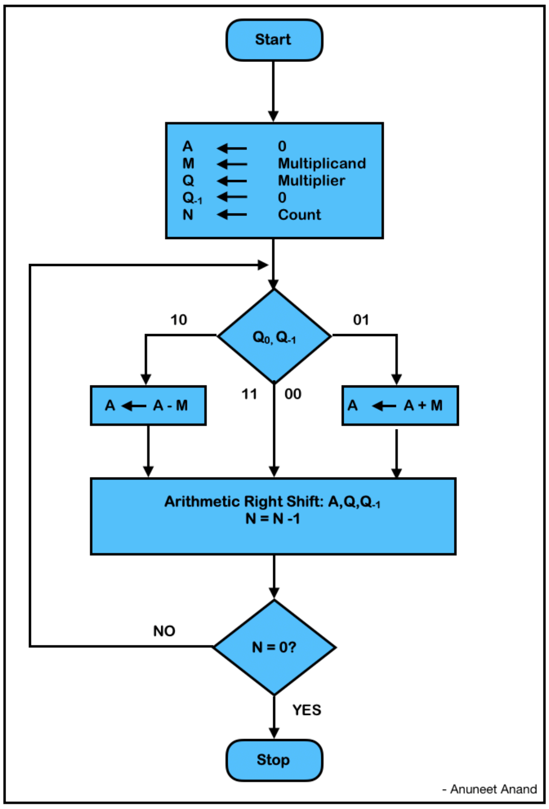
Multiplication & Division Algorithm

(Anuneet Anand & Pruthwiraj Nanda)

Booth's Multiplication Algorithm: It multiplies two signed binary numbers in two's complement notation.

1. Initialise A with 0, M with Multiplicand, Q with Multiplier, Q_{-1} with 0 and N with length of Multiplier.
2. Determine last two bits of A Q_0 Q_{-1}
   - If "10" , then subtract M from A
   - If "01" , then add M to A
   - If "00" or "11" , ignore
3. Right Shift A Q_0 Q_{-1}
4. N = N - 1
5. Repeat 2 & 3 till N becomes 0
6. A Q_0 will represent the product.



Complexity:

Consider input A,B

Let M = Max(A,B)

Let N = $\text{Log}_2(M)$ [Number Of Bits In Binary Representation Of A]

Add & Bit Manipulation Functions : O(N)

Main Algorithm : O(N)

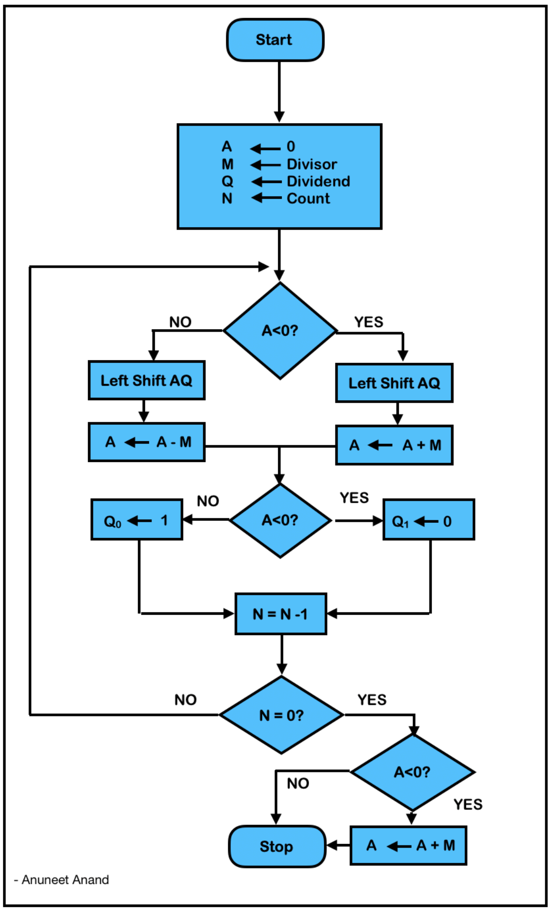
Hence Overall Complexity = $O(N^2)$ i.e. $O((\text{Log}_2(M))^2)$

Test Cases:

- Input : 2,5 Output: 10
- Input : 2,-5 Output: -10
- Input : -3,100 Output: -300
- Input : -200,-5 Output : 1000

Non-Restoring Division Algorithm: It divides two signed binary numbers and produce quotient & remainder.¶

1. Initialise A with 0, M with Divisor, Q with Dividend and N with length of Divisor.
2. Determine sign of A
   - If Positive, then Left Shift and subtract M from A
   - If Negative , then Left Shift and add A to M
3. Determine sign of A
   - If Positive, then $Q_0 = 0$
   - If Negative , then $Q_0 = 1$
4. N = N - 1
5. Repeat 2 & 3 till N becomes 0
6. Determine sign of A
   - If Negative, then Add A to M
7. Q represents Quotient and A represents Remainder
8. Assign signs to Quotient & Remainder by checking first bits of M & Q initially.



- Anuneet Anand

Complexity:

Consider input A,B

Let M = Max(A,B)

Let N = Log$_2$(M) [Number Of Bits In Binary Representation Of A]

Add & Bit Manipulation Functions : O(N)

Main Algorithm : O(N)

Hence Overall Complexity = O(N^2) i.e. O((Log$_2$(M))^2)

Test Cases:

- Input : 10,5 Output: Quotient=2, Remainder=0
- Input : 11,3 Output: Quotient=3, Remainder=2
- Input : 2,5 Output: Quotient=0, Remainder=2
- Input : -200,-5 Output : Quotient=40, Remainder=0
- Input : 7,-2 Output : Quotient=-3, Remainder=1

Assumption: Certain conventions have been followed for division of negative numbers. The remainder is considered negative when the first number is negative. Source: **https://chortle.ccsu.edu/java5/Notes/chap09B/ch09B_17.html**

Note: The algorithm has been coded in python and it is presented as a menu driven program. File : BA.py
Division by zero has been explicitly handled.

Reference:

- https://en.wikipedia.org/wiki/Booth%27s_multiplication_algorithm
- https://edurev.in/studytube/Multiplication-Algorithm--Division-Algorithm-