# Multiplication of Two Real Numbers – Halving and Doubling

## Algorithm

### Integer Multiplication

These are steps for multiplying two integer numbers:

* Find the minimum and the maximum number and save them as ‘n’ and ‘m’
* Define a variable named ‘s’ and initialize it with zero
* If ‘n’ is odd then assign the value of ‘m’ to ‘s’
* While ‘n’ is greater than one, do these:
  + Halve the value of ‘n’
  + Double the value of ‘m’
  + If ‘n’ is odd then add ‘m’ to ‘s’
* Now ‘s’ is the final result

### Real Multiplication

These are steps for multiplying two real numbers:

* If one of the numbers is zero then the result is zero
* Ignore decimal point of the numbers
* Multiply them as two integer numbers by “The Integer Multiplication Algorithm”
* The number of decimal digits of the result equals to the number of decimal digits of the first original number plus the second one
* If just one of the original numbers is negative, negate the result

## Analysis

### Integer Multiplication

|  |  |
| --- | --- |
| Step | Time Complexity |
| a = <the first integer number>;  b = <the second integer number>;  n = unsigned minimum of (a, b);  m = unsigned maximum of (a, b);  s = n mod 2 == 0 ? 0 : m; // Result | 1 |
| while (n > 1) {  n /= 2;  m += m;  if (n mod 2 == 1) {  s += m;  }  } | lg(min(a, b)) |
| END | O(lg(min(a, b))) |

### Real Multiplication

|  |  |
| --- | --- |
| Step | Time Complexity |
| a = <the first float number>;  b = <the second float number>; | 1 |
| If (a == 0 or b == 0) {  return 0;  } | 1 |
| u = unsigned a;  v = unsigned b; | 1 |
| n\_dec = number\_of\_decimal\_digits(u);  m\_dec = number\_of\_decimal\_digits(v); | log10(a) + log10(b) |
| n = ignore\_decimal\_point(u, n\_dec);  m = ignore\_decimal\_point(v, m\_dec); | 1 |
| p = integer\_multiply(n, m); | lg(min(a, b)) |
| s = p / pow(10, n\_dec, m\_dec); | 1 |
| if (a is negative) or (b is negative) {  return -s;  } else {  return s;  } | 1 |
| END | O(lg(min(a, b)) + log10(a) + log10(b)) |