Karatsuba Large Integer Multiplication:

Karatsuba algorithm is a fast multiplication algorithm that uses a divide-and-conquer approach to efficiently multiply two large integers. It was discovered by Anatolii Alexeevitch Karatsuba in 1960. The Karatsuba algorithm is a fast multiplication algorithm that uses a divide and conquer approach to multiply two numbers. The naive algorithm for multiplying two numbers has a running time of Θ(*n*^2) while this algorithm has a running time of Θ(*n*^1.585). Being able to multiply numbers quickly is very important. Computer scientists often consider multiplication to be a constant time *O*(1) operation, and this is a reasonable simplification for smaller numbers; but for larger numbers, the actual running times need to be factored in, which is *O*(*n*^2). The point of the Karatsuba algorithm is to break large numbers down into smaller numbers so that any multiplications that occur happen on smaller numbers. Karatsuba can be used to multiply numbers in all base systems (base-10, base-2, etc.).

Algorithm:

1. **Base Case:**
   - If the input integers have a small number of digits, perform a standard multiplication.
2. **Divide:**
   - Split the input integers into two halves, x = x1×*Bn*/2+*x*0 and y = *y*1×*Bn*/2+*y*0, where *B* is the base and *n* is the number of digits.
3. **Recursion:**
   - Recursively compute three products: *z*2=*x*1×*y*1,*z*0=*x*0×*y*0, and
   - *z*1=(*x*1+*x*0)×(*y*1+*y*0)−*z*2−*z*0.
4. **Combine:**
   - Calculate the final result as *result*=*z*2×*Bn*+*z*1×*Bn*/2+*z*0.

Code:

```
#include <iostream>
#include <cmath>
using namespace std;
int countDigits(int num) {
   int count = 0;
   while (num != 0) {
      num /= 10;
      count++;
   }
   return count;
}
```

```
int karatsuba(int x, int y) {
   if (x < 10 || y < 10) {
      return x * y;
   }
   int n = max(countDigits(x), countDigits(y));
   int m = (n / 2) + (n % 2);
   int high1 = x / pow(10, m);
   int low1 = x % int(pow(10, m));
   int high2 = y / pow(10, m);
   int low2 = y % int(pow(10, m));
   int z0 = karatsuba(low1, low2);
   int z1 = karatsuba((low1 + high1), (low2 + high2));
   int z2 = karatsuba(high1, high2);
   return z2 * int(pow(10, 2 * m)) + ((z1 - z2 - z0) * int(pow(10, m))) + z0;
}
int main() {
   int x = 2323;
   int y = 3232;
   int result = karatsuba(x, y);
   cout << "Result of " << x << " * " << y << " using Karatsuba algorithm: " << result << endl;
   return 0;
}
```

Output:

```
gdb.exe' '--interpreter=mi'
Result of 2323 * 3232 using Karatsuba algorithm: 7507936
```

Space complexity: _____

Justification:_____
_____

Time complexity:

Best case time complexity: _____

Justification:_____
_____

Worst case time complexity: _____

Justification:_____