

Reg.No : 24BCE0554  
 Name : Partha Pratim Gogoi  
 Topic : Karatsuba's Algorithm

### Algorithm:

```

1  num1 = ... [T.C: O(1)]
2  num2 = ... [T.C: O(1)]
3  karatsuba(num1,num2) [T.C: T(n) = O(n1.5)]
    3.1  n = max(log(num1)+1, log(num2)+1) [T.C: O(1)]
    3.2  half = n/2 [T.C: O(n)]
    3.3  p = 10half [T.C: O(n)]
    3.4  int a = num1/p [T.C: O(n)]
    3.5  int b = num1%p [T.C: O(n)]
    3.6  int c = num2/p [T.C: O(n)]
    3.7  int d = num2%p [T.C: O(n)]
    3.8  int z0 = karatsuba(a,c) [T.C: T(n) = T(n/2)]
    3.9  int z1 = karatsuba(b,d) [T.C: T(n) = T(n/2)]
    3.10 int z2 = karatsuba(a+b,c+d) [T.C: T(n) = T(n/2)]
    3.11 return z2*p2 + (z1-z2-z0)*p + z0 [T.C:O(1)]
  
```

### Time Complexity:

karatsuba(a,c)	→ T(n/2)	[halfed digits, implies 'n/2 time']
karatsuba(b,d)	→ T(n/2)	[halfed digits, implies 'n/2 time']
karatsuba(a+b,c+d)	→ T(n/2)	[halfed digits, implies 'n/2 time']
mult, div operations	→ O(n)	[defined]
max( ... )	→ O(1)	[constant time, single comparision]

Hence,

$$T.C. = [3T(n/2) + O(n)] + 50(n) + 20(1)$$

By Master's Theorem

$$T(n) = O(n^{\log_2^3}) \\ \sim O(n^{1.5})$$

$$\begin{aligned} \text{Total Time Complexity} &= O(n^{1.5}) + 50(n) + 20(1) \\ &= O(n^{1.5}) \end{aligned}$$

## Source Code:

```
#include <iostream>
#include <math.h>
using namespace std;

//////////
/// Main Logic ///
//////////
long long karatsuba(long long num1, long long num2) {
    // ----- //
    // Base Condition for smaller numbers //
    // ----- //
    if (num1<10 || num2<10) return num1*num2;

    // ----- //
    // Initialize starting variables //
    // ----- //
    int n = max(log10(num1)+1, log10(num2)+1);
    int half = n/2;
    long long p = pow(10, half);

    // ----- //
    // Split the numbers //
    // ----- //
    // First number
    int a = num1 / p;
    int b = num1 % p;
    // Second number
    int c = num2 / p;
    int d = num2 % p;

    // ----- //
    // 3 recursive multiplications //
    // ----- //
    int z0 = karatsuba(a,c);
    int z1 = karatsuba(b,d);
    int z2 = karatsuba(a+b, c+d);

    // ----- //
    // Final result //
    // ----- //
    return z2*p*p + (z1-z2-z0)*p + z0;
}

//////////
/// Driver Code ///
//////////
int main() {
    long long num1, num2;
    cout << "Enter first number: ";
    cin >> num1;
    cout << "Enter second number: ";
    cin >> num2;

    long long res = karatsuba(num1, num2);
    cout << "The result is: " << res << endl;
    return 0;
}
```

### **Sample Output:**

```
rug-arch@Oxide [Karatsuba's Algorithm]>> g++ karatsuba.cpp
rug-arch@Oxide [Karatsuba's Algorithm]>> ./a.out
Enter first number: 14241
Enter second number: 1235
The result is: 469610834
```