

```
In [1]: import numpy as np
        from itertools import zip_longest
```

algorithm

```
In [2]: def add(x, y):
        z, carry = [], 0

        for r, s in zip_longest(x, y, fillvalue=0):
            t = r + s + carry
            carry = t // 10
            z.append(t % 10)
        if carry:
            z.append(carry)

        return z
```

```
In [3]: def sub(x, y):
        z, carry = [], 0

        for r, s in zip_longest(x, y, fillvalue=0):
            t = r - s + carry
            carry = t // 10
            z.append(t % 10)

        return z
```

```
In [4]: def karatsuba(x, y):
        # ensure same length
        while len(x) < len(y):
            x.append(0)
        while len(x) > len(y):
            y.append(0)

        # length and split
        n = len(x)
        n_2 = (n + 1) >> 1

        # trivial case
        if n == 1:
            return add([x[0] * y[0]], [])

        # split
        x0, x1 = x[:n_2], x[n_2:]
        y0, y1 = y[:n_2], y[n_2:]

        # karatsuba algorithm
        z0 = karatsuba(x0, y0)
        z1 = karatsuba(x1, y1)
        z2 = karatsuba(add(x0, x1), add(y0, y1))
        z2 = sub(sub(z2, z0), z1)
```

```
z = add(z0, [0] * (n_2 << 1) + z1)
z = add(z, [0] * n_2 + z2)

return z
```

run

```
In [5]: def mult(x, y):
        print(x, '*', y, '=', int(x) * int(y), end=' = ')

        x = list(map(int, reversed(x)))
        y = list(map(int, reversed(y)))
        z = karatsuba(x, y)

        print(''.join(map(str, reversed(z))))
```

```
In [6]: mult('1234', '4321')

1234 * 4321 = 5332114 = 5332114
```

```
In [7]: mult('5678', '8765')

5678 * 8765 = 49767670 = 49767670
```

```
In [8]: mult('9999', '9999')

9999 * 9999 = 99980001 = 99980001
```

```
In [9]: for _ in range(30):
        n = np.random.randint(1, 11)
        x = ''.join(map(str, np.random.randint(0, 10, n)))
        y = ''.join(map(str, np.random.randint(0, 10, n)))
        mult(x, y)

60504 * 36056 = 2181532224 = 2181532224
7644 * 2034 = 15547896 = 15547896
2 * 1 = 2 = 2
939 * 700 = 657300 = 657300
977707258 * 389036934 = 380364234001866972 = 380364234001866972
1079459668 * 7762768164 = 8379595145072409552 = 8379595145072409552
4609807 * 2350979 = 10837559451053 = 10837559451053
36740 * 97490 = 3581782600 = 3581782600
29 * 19 = 551 = 0551
913789 * 733694 = 670441506566 = 670441506566
476646777 * 451303071 = 215112154242352167 = 215112154242352167
23369 * 43069 = 1006479461 = 1006479461
9185982 * 3983922 = 36596235781404 = 036596235781404
7806584211 * 8415537629 = 65696603181627775719 = 65696603181627775719
317 * 106 = 33602 = 33602
507729648 * 898501571 = 456195886371277008 = 456195886371277008
780843 * 778950 = 608237654850 = 608237654850
61 * 79 = 4819 = 4819
310094 * 443993 = 137679565342 = 137679565342
1564 * 7634 = 11939576 = 11939576
746602083 * 909270015 = 678862887208441245 = 678862887208441245
```

