Design and Analysis of Algorithms

- ***Course Code***: BCSE304L
- ***Course Type***: Theory (ETH)
- ***Slot***: A1+TA1 & & A2+TA2
- ***Class ID***: VL2023240500901
  VL2023240500902

A1+TA1

Day	Start	End
Monday	08:00	08:50
Wednesday	09:00	09:50
Friday	10:00	10:50

A2+TA2

Day	Start	End
Monday	14:00	14:50
Wednesday	15:00	15:50
Friday	16:00	16:50

Syllabus- Module 1

Module:1	Design Paradigms: Greedy, Divide and Conquer Techniques	6 hours

Overview and Importance of Algorithms - Stages of algorithm development: Describing the problem, Identifying a suitable technique, Design of an algorithm, Derive Time Complexity, Proof of Correctness of the algorithm, Illustration of Design Stages -

Greedy techniques: Fractional Knapsack Problem, and Huffman coding

Divide and Conquer: Maximum Subarray, Karatsuba faster integer multiplication algorithm.

Karatsuba faster integer multiplication algorithm

Definition:

The Karatsuba algorithm is a high-performance multiplication algorithm designed for efficiently multiplying two large integers.

Anatolii Alexeevitch Karatsuba discovered this algorithm in 1960, and it is renowned for its efficiency when compared to traditional multiplication algorithms.

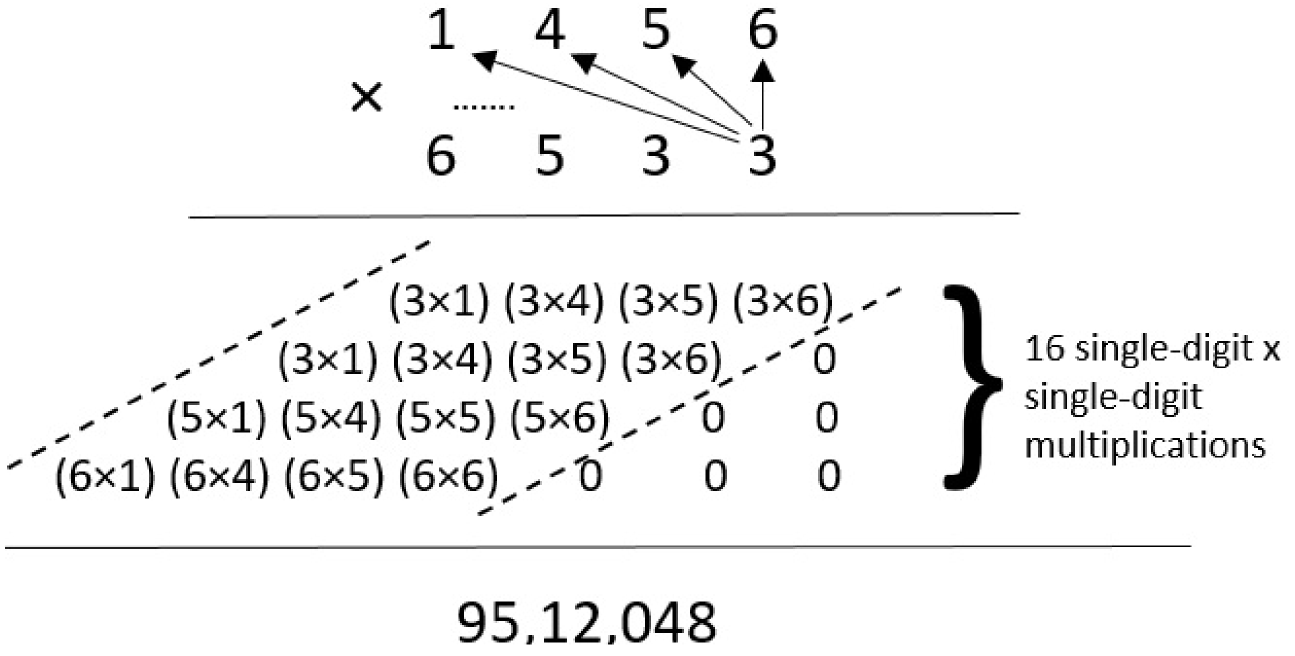
Use:

Utilized by systems to perform rapid multiplication of two n-digit numbers, the Karatsuba algorithm significantly reduces the computation time required by the system compiler compared to the time taken by standard multiplication methods.

Karatsuba faster integer multiplication algorithm

Time Complexity of normal multiplication in Brute Force Approach

Consider two 4-digit integers: **1456** and **6533**, and find the product using Naive approach.
So, **1456 × 6533 =?**



In the naive multiplication method, if both numbers have n digits, there are n x n single-digit × single-digit multiplications being performed. Thus, the time complexity of this approach is O(n^2) since it takes n^2 steps to calculate the final product.

Karatsuba faster integer multiplication algorithm

in **Divide and conquer technique**

If x and y are two number of Create the following three subproblems where H represents the high bits of the number and L represents the lower bits:

- Three subproblems:
  - $a = xH \cdot yH$ ➔ Multiplying the high bits of both numbers.
  - $d = xL \cdot yL$ ➔ Multiplying the low bits of both numbers
  - $e = (xH + xL)(yH + yL) - a - d$ ➔ Calculating the cross product of the sum of high and low bits, then subtracting the results of the previous multiplications.

Then The product *xy* is expressed as: $xy = a.r^n + e \cdot r^{(n/2)} + d$

Karatsuba faster integer multiplication algorithm in Divide and conquer technique

- $a = x_H y_H$
- $d = x_L y_L$
- $e = (x_H + x_L)(y_H + y_L) - a - d$
- $xy = ab^n + eb^{\frac{n}{2}} + d.$

Step-by-step breakdown of the Karatsuba algorithm applied to the multiplication of 1234 and 4321.

Given Numbers:

- $x = 1234$ and $y = 4321$
- $x_{\mathrm{H}} = 12, y_{\mathrm{H}} = 43$
- $x_{\mathrm{L}} = 34, y_{\mathrm{L}} = 21$

Subproblems:

- $a_1 = 12 \times 43$
- $d_1 = 34 \times 21$
- $e_1 = (12 + 34) \times (43 + 21) - a_1 - d_1$

Recursion for d_1: 34×21

- $a_2 = 3 \times 2 = 6$
- $d_2 = 4 \times 1 = 4$
- $e_2 = (3 + 4) \times (2 + 1) - a_2 - d_2 = 11$
- Answer: $6 \times 10^2 + 11 \times 10 + 4 = 714$

Recursion for a_1: 12×43

- $a_2 = 1 \times 4 = 4$
- $d_2 = 2 \times 3 = 6$
- $e_2 = (1 + 2) \times (4 + 3) - a_2 - d_2 = 11$
- Answer: $4 \times 10^2 + 11 \times 10 + 6 = 516$

Recursion for e_1: 46×64

- $a_2 = 4 \times 6 = 24$
- $d_2 = 6 \times 4 = 24$
- $e_2 = (4 + 6) \times (6 + 4) - a_2 - d_2 = 52$
- Answer: $24 \times 10^2 + 52 \times 10 + 24 - 714 - 516 = 1714$

Final Answer:

- $1234 \times 4321 = 516 \times 10^4 + 1714 \times 10^2 + 714 = 5,332,114$

Karatsuba faster integer multiplication algorithm

Time Complexity in Divide and conquer technique

Recurrence Relation

T(n) = 3 T(n/2) + O(n)

T(n) = O(nlog 3) = O(n^(1.584…))

- $a = x_H y_H$
- $d = x_L y_L$
- $e = (x_H + x_L)(y_H + y_L) - a - d$
- $xy = ab^n + eb^{\frac{n}{2}} + d.$

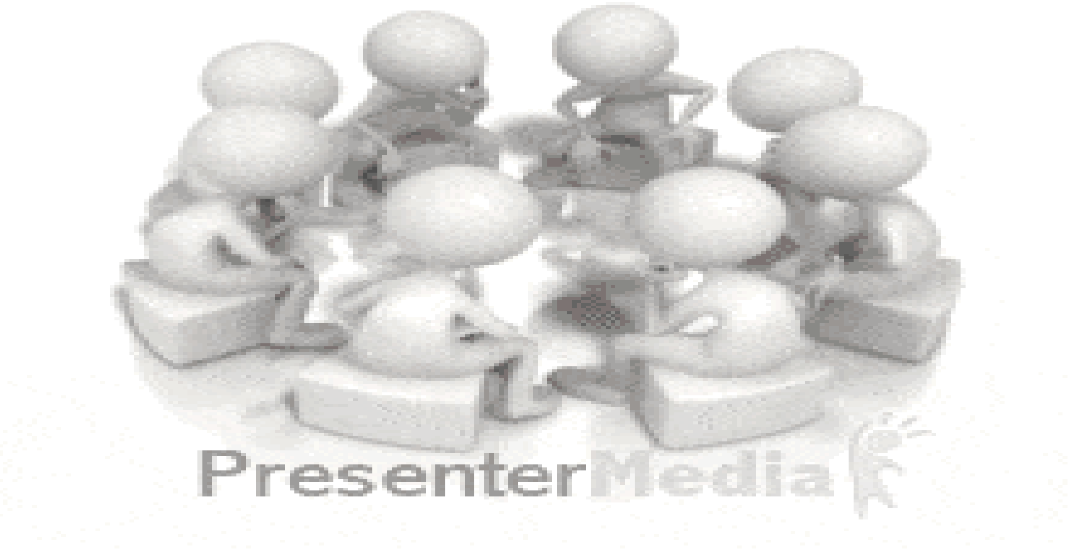
```
Algorithm karatsuba_multiply(x, y):
    if x < 10 or y < 10:
        return x * y  // Base case: Use standard multiplication for small numbers
    n = max(length(x), length(y))
    m = n / 2  // Middle index for splitting

    // Split the input integers into halves
    xH, xL = split(x, m)
    yH, yL = split(y, m)

    // Recursive calls
    a = karatsuba_multiply(xH, yH)
    d = karatsuba_multiply(xL, yL)
    e = karatsuba_multiply(xH + xL, yH + yL) - a - d
    // Combine the results
    result = a * 10^(2*m) + e * 10^m + d
    return result
```

```
// Example usage:
result = karatsuba_multiply(1234, 4321)
print(result)
```


Any
Question


PresenterMedia


Thank You!


FOR YOUR
ATTENTION