Integer Multiplication

Given TWO LARGE positive integers of ***N*** digits/each. Each integer is stored in 1D array. Implement an **efficient algorithm** based on **Karatsuba’s** method to multiply them?

**NOTES:**

* ***N*** is power of 2 (i.e. 2, 4, 8, 16, 32… 2i)
* Result MUST be stored in **2×N** digits (left padded by 0’s if necessary)
* **Least significant** digit is stored at **index 0** while most significant is stored at index N-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Index** | **N – 1** | **…** | **1** | **0** |
| **Digit** | Most signif. digit | … | 2nd digit | Least signif. digit |

## Complexity

The complexity of your algorithm should be **less than O(N2)**

## Evaluation

|  |  |  |
| --- | --- | --- |
| **Sample Cases**  (Correctness) | **UNSEEN Large Cases**  (Efficiency) | **Total** |
| 2 Marks | 6 Marks | 8 MARKS |

## Bonus & Competition#2

|  |  |  |
| --- | --- | --- |
|  | **Criteria** | **BONUS** |
| **Vs.** **Naïve**  (on Large Cases) | **Just Faster** | +1 Mark |
| **1x Faster** | +3 Marks |
| **[N]x Faster** | +**[N]x2** Marks |
| **TOP5** | Correct & Speed | 2~4 Marks |

Function: **Implement it!**

static public byte[] IntegerMultiply(byte[] X, byte[] Y, int N)

IntegerMultiplication.cs includes this method.

## Examples

|  |  |
| --- | --- |
| **EX#1** | **EX#2** |
| **X:**   |  |  |  |  | | --- | --- | --- | --- | | 9 | 9 | 9 | 9 |   **Y:**   |  |  |  |  | | --- | --- | --- | --- | | 9 | 9 | 9 | 9 | | **X:**   |  |  |  |  | | --- | --- | --- | --- | | 0 | 2 | 2 | 2 |   **Y:**   |  |  |  |  | | --- | --- | --- | --- | | 0 | 0 | 1 | 1 | |
| **Res:**   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** | | 9 | 9 | 9 | 8 | 0 | 0 | 0 | 1 | | **Res:**   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** | | 0 | 0 | 0 | 0 | 2 | 4 | 4 | 2 | |

# C# Help

## Getting the size of 1D array

int size = array1D.GetLength(0);

## Getting the size of 2D array

int size1 = array2D.GetLength(0);

int size2 = array2D.GetLength(1);

## Creating 1D array

int [] array1D = new int [size]

## Creating 2D array

int [,] array2D = new int [size1, size2]

## Sorting single array

Sort the given array "items" in ascending order

Array.Sort(items);

## Sorting parallel arrays

Sort the first array "master" and re-order the 2nd array "slave" according to this sorting

Array.Sort(master, slave);