

Sort - and - counting

// finds total no. of inversion using divide and conquer
 // input: Array A of n distinct integers
 // output: sorted array B with the same integers
 // and no. of inversions of A

if $n=0$ or $n=1$ then

return (A, 0)

else

(C, leftinv) = sort - and - counting (first half of A)

(D, rightinv) = sort - and - counting (second half of A)

(B, splitinv) = merge - and - countsplition (C, D)

return (B, leftinv + rightinv + splitinv)

merge - and - countsplition

// input: sorted arrays C and D (length $n/2$ each)

// output: sorted array B and no. of splitinv

i = 1, j = 1, splitinv = 0

for K = 1 to n

if $C[i] < D[j]$

B[K] = C[i]

i = i + 1

else

B[K] = D[j]

j = j + 1

splitinv = splitinv + ($\frac{n}{2} - i + 1$)

return (B, splitinv)

Brute force

// finds no. of inversion in Array A
 // input: Array A of n distinct integers
 // output: The no. of inversions of A

numinv = 0

for i = 1 to n-1

if A[i] > A[j]

numinv = numinv + 1

return numinv

Karatsuba

// finds multiplication of a and b

// input: two n-digit positive integers x and y

// output: The product of x.y

if n = 1

compute x.y in one step and return result

else

a, b = first and second half of x

c, d = first and second half of y

compute p = a+b and q = c+d using grade school add

recursively compute ac = a.c, bd = b.d and pq = p.q

compute adbc = pq - ac - bd using grade school add

compute $10^n \cdot ac + 10^{n/2} \cdot adbc + bd$ using grade school add and return result

Karatsuba

$$T(n) = 3T\left(\frac{n}{2}\right) + cn$$

$$a = 3$$

$$b = 2$$

$$d = 1$$

$$a > b^d$$

$$3 > 2^1$$

$$3 > 2$$

using master theorem $\Rightarrow n^{\log_b a} = n^{\log_2 3} = n^{1.59}$

counting divide: The divide step computes the middle of the subarray which takes constant time,
 $D(n) = O(1)$

conquer: we recursively compute 2 subproblems each of size $n/2$ which contributes
 $2T(n/2)$

combine: $C(n) = O(n)$

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

$$a=2, b=2, d=1$$

$$a = b^n$$

$$2 = 2^1$$

using master theorem: $n^d \log n = n \log n$

Brute force counting

time 2 for loops nested are present the time complexity will be $O(n^2)$

countinv

negative TC:

1] []

Output: The list of course code is empty so inversion
cant be found

2] [-2, -1, -3]

Output: inversion cant be found since course code
cant be negative

3] ['a', 1, 4]

Output: inversion cant be found since letters are
present

4] [True, 3, 4, 5]

Output: inversion cant be found since boolean value present

5] [-1, 'a', 3, 4]

Output: inversion cant be found since both negative
value and string is present

Karatsuba

4

Brute force

Karatsuba

A

brute force

negative TC

1] $a = "mihir"$ $b = 1375$

inversions

Output: multiplication not possible between
integer and string text2] $a = "1234"$ $b = 1230$ Output: multiplication not possible between
string and integer3] $a = 123.46$ $b = 230$ Output: multiplication of float with integer not
possible4] $a = \text{True}$ $b = 231$ Output: multiplication between boolean value and
integer not possible5] $a = 1 + 2j$ $b = 5678$ Output: multiplication between complex no. and
integer not possible

Conclusion :

In this experiment I learnt how to code the programs to find both count inversions and Karatsuba this allowed me to gain practical experience. I considered a good set of positive and negative test cases and at each step displayed the appropriate message to the user whether the output exists or not.