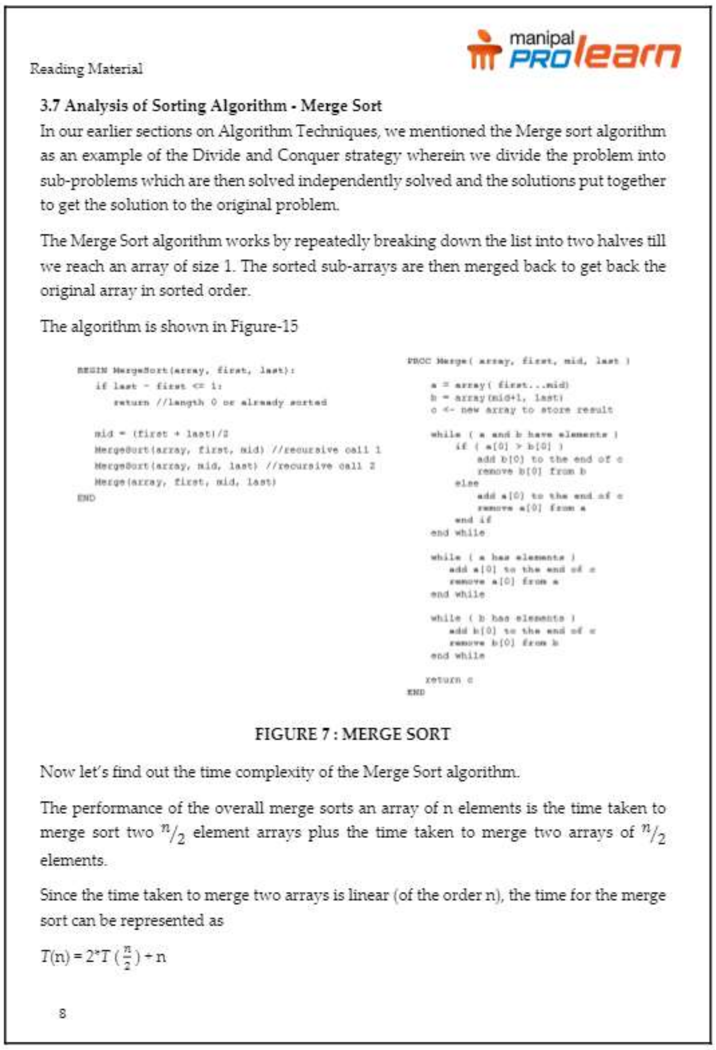
(/)

S1_01 Data Structures And Algorithms : Subject Content

Course Menu

Zoom Size: 100%

< Previous Page: 8 / 10 Next >



3.7 Analysis of Sorting Algorithm - Merge Sort

In our earlier sections on Algorithm Techniques, we mentioned the Merge sort algorithm as an example of the Divide and Conquer strategy wherein we divide the problem into sub-problems which are then solved independently solved and the solutions put together to get the solution to the original problem.

The Merge Sort algorithm works by repeatedly breaking down the list into two halves till we reach an array of size 1. The sorted sub-arrays are then merged back to get back the original array in sorted order.

The algorithm is shown in Figure-15

```
BEGIN MergeSort(array, first, last):
   if last - first <= 1:
      return //length 0 or already sorted

   mid = (first + last)/2
   MergeSort(array, first, mid) //recursive call 1
   MergeSort(array, mid, last) //recursive call 2
   Merge(array, first, mid, last)
END
```

```
PROC Merge( array, first, mid, last )

   a = array( first...mid)
   b = array(mid+1, last)
   c <- new array to store result

   while ( a and b have elements )
       if ( a[0] > b[0] )
           add b[0] to the end of c
           remove b[0] from b
       else
           add a[0] to the end of c
           remove a[0] from a
       end if
   end while

   while ( a has elements )
      add a[0] to the end of c
      remove a[0] from a
   end while

   while ( b has elements )
      add b[0] to the end of c
      remove b[0] from b
   end while

  return c
END
```

FIGURE 7 : MERGE SORT

Now let's find out the time complexity of the Merge Sort algorithm.

The performance of the overall merge sorts an array of n elements is the time taken to merge sort two $^{n}/_{2}$ element arrays plus the time taken to merge two arrays of $^{n}/_{2}$ elements.

Since the time taken to merge two arrays is linear (of the order n), the time for the merge sort can be represented as

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

Views : 15686

230 0

< qsort() Library Function Demo (/?q=MULNCourseBook/listDetailedCBLearningContents/178099/cidb/full/view/153784/431550/431535/39731)