DIVIDE AND CONQUER

INTRODUCTION TO ALGORITHMS

Made by -
Shivam Joshi

INTRODUCTION

In this presentation, we are going to discuss my problem statement, how Divide and Conquer technique is helpful and how we can solve the problem with the DAC technique approach.

Divide-and-conquer, breaks a problem into subproblems that are similar to the original problem, recursively solves the subproblems, and finally combines the solutions to the subproblems to solve the original problem. Because divide-and-conquer solves subproblems recursively, each subproblem must be smaller than the original problem, and there must be a base case for subproblems.

Technique of DAC

This technique can be divided into the following three parts:

1. **Divide:** This involves dividing the problem into some sub problem.
2. **Conquer:** Sub problem by calling recursively until sub problem solved.
3. **Combine:** The Sub problem Solved so that we will get find problem solution

Problem statement

Simplified problem statement -

Implement merge sort

Original statement -

Show the marks of students in your class in ascending order.

Approach -

We are now going to implement merge sort for the given statement

Solution

Because we're using divide-and-conquer to sort, we need to decide what our subproblems are going to look like. The full problem is to sort an entire array. Let's say that a subproblem is to sort a subarray. In particular, we'll think of a subproblem as sorting the subarray starting at index p and going through index r. It will be convenient to have a notation for a subarray, so let's say that array[p..r] denotes this subarray of array.In terms of our notation, for an array of n elements, we can say that the original problem is to sort array[0..n-1].

Here's how merge sort uses divide-and-conquer:

1. **Divide** by finding the number q of the position midway between p and r. Do this step the same way we found the midpoint in binary search: add p and r, divide by 2, and round down.
2. **Conquer** by recursively sorting the subarrays in each of the two subproblems created by the divide step. That is, recursively sort the subarray `array[p..q]` and recursively sort the subarray `array[q+1..r]`.
3. **Combine** by merging the two sorted subarrays back into the single sorted subarray `array[p..r]`.

Working of DAC in merge sort