Divide & Conquer Algorithms

Divide and Conquer Algorithms

- breaks the input to several parts
- solves the problem in each part (usually recursively)
- combines the solutions to these subproblems into an overall solution

MergeSort Algorithm

- Divide the array into two equal lists
- Sort each sub-list recursively
- merge the two sorted sub-lists to become one sorted list

MergeSort Algorithm

MergeSort A[1,...n]

```
if n=1, return A

A_1 = MergeSort A[1,…,n/2]

A_2 = MergeSort A[n/2+1,…,n]

A = Merge(A_1, A_2)

return A
```

MergeSort Algorithm

Merge(A,B)

```
p=1,q=1

while p ≤ n and q ≤ n

    if A[p] < B[q]

        add A[p] to C; p++ ;

    else

        add B[q] to C; q++ ;

if p > n

    Add the remaining elements of A to C

else

    Add the remaining elements of B to C
```

MergeSort Algorithm

Correctness of MergeSort :

MergeSort Algorithm

Correctness of MergeSort :

- By induction on the size of the array
- Correctness of Merge()
    - After the ith iteration, C contains the ith smallest element in the ith position - By induction on i.

MergeSort Algorithm

Running Time of MergeSort :

MergeSort Algorithm

Running Time of MergeSort :

Time taken for Merge() -

- For every iteration of the while loop, an element is added to the merged list.
- The total time is of the order of size of the merged list i.e, $O(n)$

- $T(n)$ - time for MergeSort[1, ..., n]
- $T(n) = 2T(n/2) + O(n)$

Solving Recurrences

Unroll the recurrence :

Recursion Tree :

Substitution Method :

Master Method :

To solve recurrences of the form

$$T(n) = aT(\frac{n}{b}) + f(n)$$

where $a \geq 1, b > 1$, $f(n)$ is asymptotically positive.

Master Method :

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some $\epsilon > 0$, then

$T(n) = \Theta(n^{\log_b a})$

Master Method :

2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \log n)$

Master Method :

3. If $f(n) = \Omega(n^{\log_b a+\epsilon})$ for some $\epsilon > 0$ and

if $af(\frac{n}{b}) \leq cf(n)$ for some $c < 1$and all large n, then

$T(n) = \Theta(f(n))$