Divide-and-Conquer

What is it?

- Divide the big problem into small sub-problems.
- Solve the sub-problems using recursion until the particular sub-problem is solved.
- combine the sub-problems so that we can be able to get the final solution.

Key point: If the problem is big, then only divide the problem otherwise, no need to divide the problem into subproblem.

```
DAC(array, i , j):
  // If the array size is small then directly return the solution
  if small(array, i, j):
        return (solution(array, i,j))

  // if the array size is large:
  else:
        m = Divide(array, i, j)
        b = DAC(array, i, m)
        c = DAC(array, m+1,j)
        return combine(b,c)
```

Time Complexity:

- if n is small

$$T(n) = O(1)$$

- If n is large:

$$T(n) = f1(n) + T(n/2) + T(n/2) + f2(n)$$

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

- F1(n) : time to divide
- T(n/2) : time to solve the subproblem
- F2(n) : time to combine

Application of Divide and Conquer

- Binary Search
- Merge Sort
- Quick Sort
- Selection Procedure
- Finding maxima and minima in the given array of elements
- Finding number of inversions
- finding power of elements
- strassens matrix multiplication and so on