Divide & ***Conquer***

➷ Introduction:

- It is primarily a recursive method.

- The divide-and-conquer strategy consists:

    ✓ in breaking a problem into simpler subproblems of the same type,

    ✓ next to solve these subproblems,

    ✓ and finally to merge the obtained results into a solution to the problem.

➷ General approach:

```
Procedure or function div_conq
Begin
        If the input is small then
              solve directly and return;
        else begin
                    Split the input into 2 or more parts.
                    Solve the problem for those smaller
                    parts.
                    Combine the solutions and return;
              end;
end;
```

- Time complexity:

  i_s: size of the small input.
  r: number of times the input is split.
  n_i: the size of each part.

  $T(n)= C \quad \text{if } n= i_s$

  $T(n)= T_{split} + T_{combine} + \sum T(n_i) \quad \text{if } n \neq i_s$

  Where

  - ✓ $T_{combine}$ is the time to combine the solutions
  - ✓ $T(n_i)$ is the time needed by each r part, and
  - ✓ T_{split} is time to split the problem into r parts.

- Examples

  - Maximum of a sequence of n elements.
  - Closest pair: Given a set of n points (x_i, y_i) the problem asks what is the distance between the two closest points.
    - A brute force approach in which the distances for all the pairs are measured takes $O(n^2)$ time.
    - A divide and conquer solution takes $T(n) = O(n \log n)$.
  - Binary search.
  - Merge sort.
  - Quick sort.
  - Selection.
  - Matrix multiplication.

➯ **Binary search:**

- Search trees:

  - ✓ Data structures to support dynamic-set operations
  - ✓ Operations: minimum, maximum, insert, delete, predecessor, successor, and search.

  - ✓ <u>Predecessor (S,x):</u> An operation that, given an element x whose key is from an ordered set S, returns the next smaller element in S, or NIL if x is the minimum element.
  - ✓ <u>Successor (S,x):</u> An operation that, given an element x whose key is from an ordered set S, returns the next maximum element in S, or NIL if x is the maximum element.

- <u>Binary search trees:</u>

  - ✓ binary trees.

  - ✓ All the operations takes O (log n) in a randomly built binary search tree.

  - ✓ Recursive function for searching operation:

    Input: given a key, k, and the root of BST T.

    Output: returns the pointer of the node containing k.

```
Function search (T,k):
Begin
        If  T=nil  or  k= data (T)
        Then    return T;
        Else begin
                        If   k < data (T)
                        then  return  search (left (T),k )
                        else  return  search (right (T),k )
                        endif;
                end;
        Endif
End.
```

Analysis:

$T(n) = T(n/2) + C \qquad \text{if } n \neq 1$

$T(n) = C$

$\Rightarrow$ **O (log n)**

➯ **Merge Sort**

- Worst - case time complexity is O (n log n)
- Procedure Merge sort (L,h)

  ∗ A(L,h) is a global array entring h-l+1 ≥ 0

  ∗ Values which represent the elements to be pertril

```
Integer   mid;
Begin
  If  l < h
 Then    begin
            mid =   (L,h)/2 ;
            call mergesort ( l,mid);
            call mergesort (mid+1,h);
            call  merge (l,mid,h);
         end
  endif
end.
```

- Analysis:

$T(n)= 2T(n/2)+ n \qquad n>1$

$T(n)= 1 \qquad n=1$

<u>T(n)= O (n log n).</u>

➪ **Matrix multiplication**

- Brute force method "direct" takes O (n^3)

  ✓ Algorithm:

```
for i:= 1 to N do
  for j:=1 to N do
     C[i,j] := 0;
     for k := 1 to N do
        C[i,j] := C[i,j] + A[i,k] * B[k,j]
```

  ✓ Analysis: $O(N^3)$ multiplications

- Simple Divide and conquer algorithm:

  ✓ Decompose the matrix into 4 submatrix

  ✓ Analysis:

$$T(n)=8\,T(n/2)+Cn2 \qquad n \neq 2$$
$$T(n)= b \qquad n = 2$$

Because

$$C_{11} = A_{11}\,B_{11} + A_{12}\,B_{21}$$
$$C_{12} = A_{11}\,B_{12} + A_{12}\,B_{22}$$
$$C_{21} = A_{21}\,B_{11} + A_{22}\,B_{21}$$
$$C_{22} = A_{21}\,B_{12} + A_{22}\,B_{22}$$

- Divide and conquer using STRASSEN'S METHOD:

  - ✓ Needs 7 multiplications & 18 additions.

  - ✓ First compute the following:

$$P= (A11+A22)(B11+B22)$$
$$Q= (A21+A22) B11$$
$$R= A11 (B12-B22)$$
$$S= A22 (B21-B11)$$
$$T= (A11+A12) B22$$
$$U=(A21-A11)(B11+B12)$$
$$V=(A12-A22)(B21+B22)$$

$$C11= P+S-T+V$$
$$C12= R+T$$
$$C21= Q+S$$
$$C22= P+R-Q+U$$

- ✓ Analysis: Assume that $n = 2^k$

$$T(n) = b \qquad n=2$$

$$T(n)= 7T(n/2)+a*n^2 \qquad n>2$$

$$\Rightarrow T(n)= O(n^{\log_2 7})= O(n^{2.81})$$