Mathematical Analysis for Non-recursive Alg

→ It executed only once to solve the Problems.

General plan:-

(1) Decide on a I/P size.
(2) Identify the basic operations
(3) Check the no. of times basic operation is executed depend only on the size of I/P
(4). Sum No. of time basic operation is executed.
(5). Simplify the sum of using standared formula & Rules.

Ex1:

Finding the value of the largest element in a list of 'n' numbers

Alg:

Algorithm max element (A[0.....n-1])

//problem Description: This alg. is for finding max-element in array.

//I/P : Array A[0.....n-1]

//O/P : returns the largest elements.

```
Max ← A[0]
for i ← 1 to n-1 do
{   If (A[i] > max) then
    max ← A[i]
    return max;
}
```

Mathematical Analysis:

Step1: The I/P size is 'n', the total no. of elements in array.

Step2: The basic operation is comparision in loop for finding large values.

Step3: The comparison is executed on each repetation of the loop. As the comparison made for each value of 'n'. There is no need to find best Avg, worst case.

Step4: Let C(n) be the no. of times comparion is executed. The alg makes comparion each time the loop executes

Hence, for i=1, to n-1 times, the comparion made,

$$C(n) = \sum_{i=1}^{n-1} 1$$

→ one comparion made for each value of i.

Step5: Simplify:

$$C(n) = \sum_{i=1}^{n-1} 1$$

$$C(n) = n-1 \in \Theta(n).$$

Thus the time complexity of finding maximum elements in an array is $\Theta(n)$.