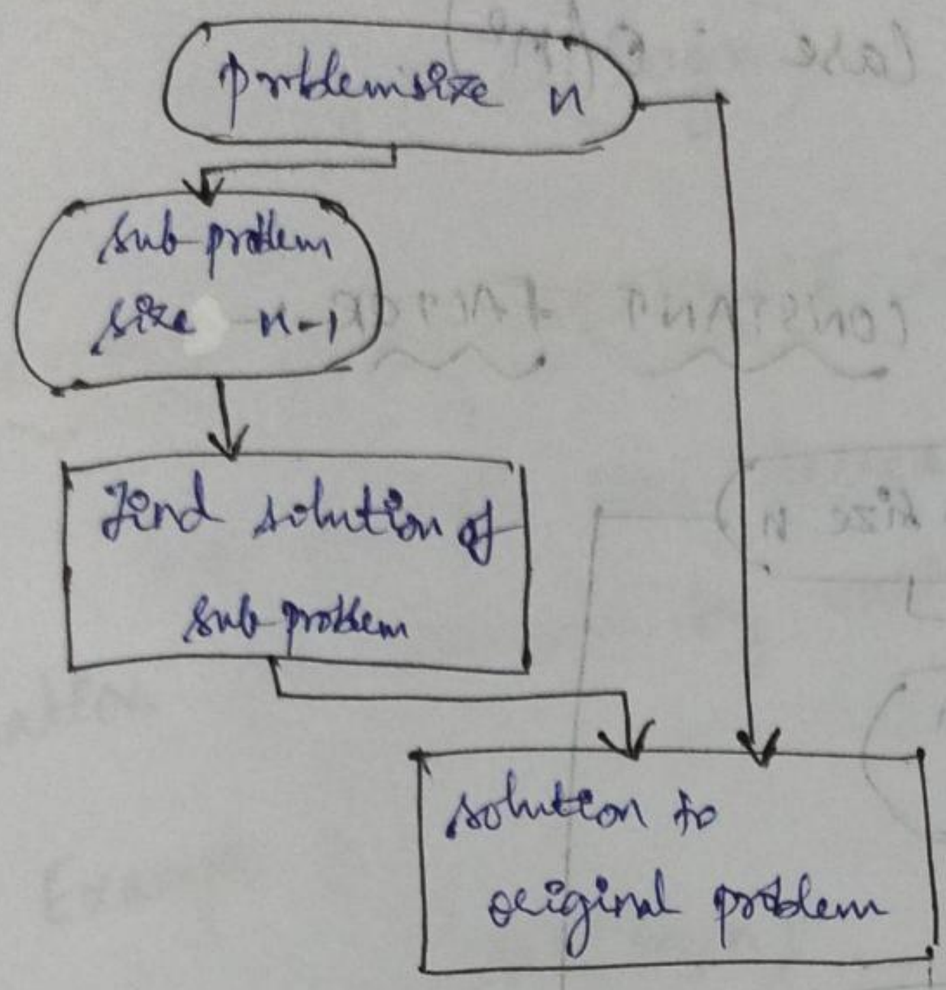
## DECREASE BY CONSTANT :-

16CSR89
M. karthik




Example :-

Insertion sort :-

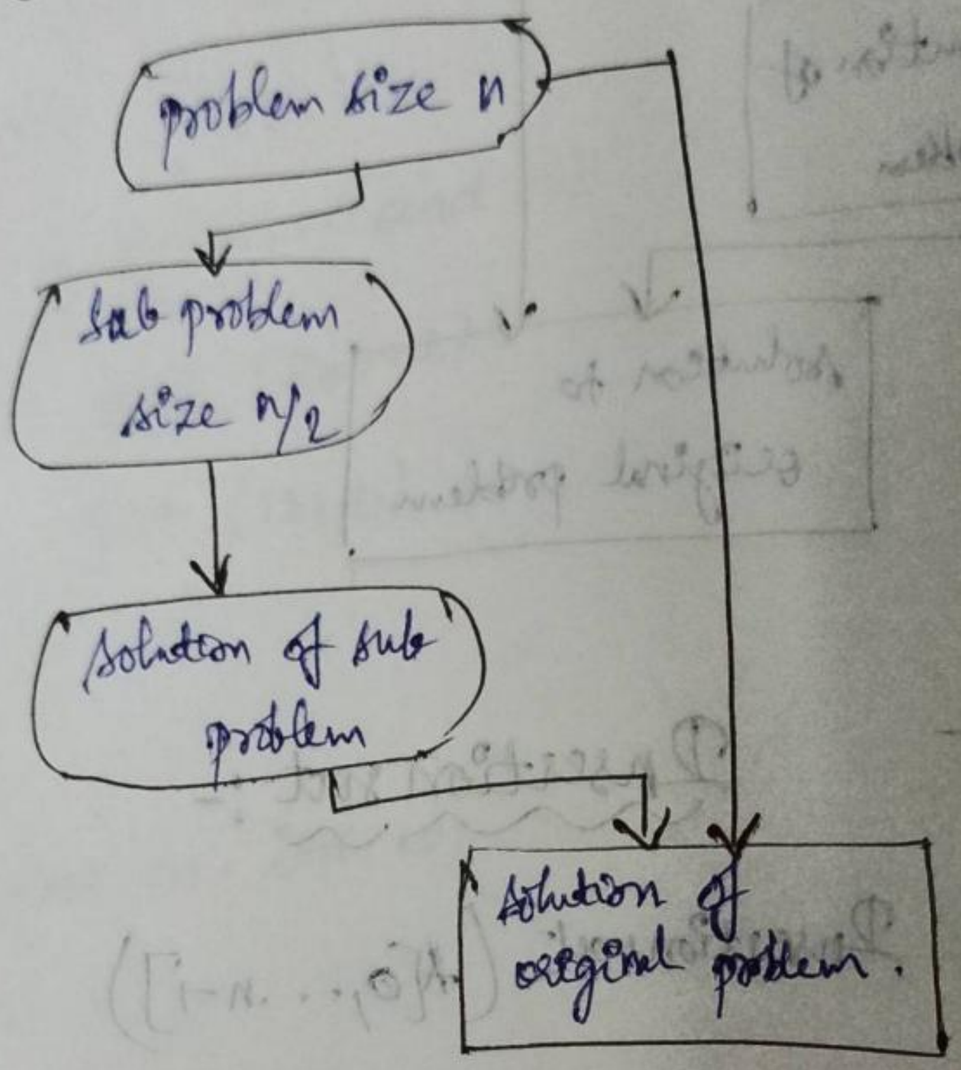
```
Alg Insertionsort (A[0,...n-1])
{
for i=1 to n-1
{
    v = A[i]; j = i-1;
    while (j ≥ 0 and A[j] > v)
    {
        A[j+1] = A[j]
        j = j-1
    }
    A[j+1] = v
}
```

$$T(n) = \sum_{i=1}^{n-1} \left[ \sum_{j=0}^{i-1} 1 \right] = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \Rightarrow \in O(n^2)$$

Best case : $O(n)$

Avg, Worst case : $O(n^2)$.

DECREASE BY CONSTANT FACTOR :-




Example :-

Binary Search :-

```
Alg binsearch (A[0..n-1], key)
{
    l=0
    r=n-1
    while (l ≤ r)
    {
        m = (l+r)/2
        if (A[m]==key)
            return m
        else if key < A[m]
            r=m-1
        else
            l=m+1
    }
    return -1
}
```

Execution Time :-

$T(n) \in O(\log n)$.

VARIABLE SIZE DECREASE :-

In variable size decrease, each and every iteration is decrease by variable size.

Example :

gcd(m, n).

```
gcd (m, n)
{
    if (n==0)
        return m;
    else
    {
        t=m%n;
        gcd(n,t);
    }
}
```

gcd(60, 24) = gcd(24, 12) = gcd(12, 0)

## TRANSFORM AND CONQUER :-

convert the problem into another type and solve.

Example :-

Heap Sort

Types :-

* Instance Simplification.
  - uniqueness
  - mode
  - binarysearch

  } → These are all applications of presorting.

* representation change.
  Ex: Heap sort.
* problem reduction.

Mode :-

Repeated no's more times called as mode.

Algorithm :-

```
Alg_mode (A[...])
{
  count=0; Num=A[0]; count1=1;
  for(i=0 to n-1)
  {
    if(A[i]==A[i+1])
      count1++;
    else if(count1 > count)
    {
      count = count1;
      Num = A[i];
      count1=1; } else count1=1;
```

```
    }
      display (count, Num);
  }
```

Example :-

Uniqueness :-

```
Alg unique (A[0,... n-1])
{
  for i=0 to n-2
  {
    if (a[i] == a[i+1])
    return false
    }
    return true
  }
```