Merge Sort :

```
Algorithm MergeSort(l, h) ← T(n)
{
    if (l < h)
    {
        mid = (l+h)/2 ;          — 1
        MergeSort(l, mid) ;      — T(n/2)
        MergeSort(mid+1, h) ;    — T(n/2)
        Merge(l, mid, h) ;       — n
    }
}
```

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

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

$a = 2$
$b = 2$
$k = 1$
$p = 0$

$a = b^k$, $p > -1$

$\Rightarrow T(n) = n^{\log_b^a} \log_b^{p+1} n$

$= n \log n$

$$T(n) = aT\left(\frac{n}{b}\right) + \Theta(n^k \log_b^p n)$$

$a > b^k \Rightarrow T(n) = \Theta(n^{\log_b^a})$

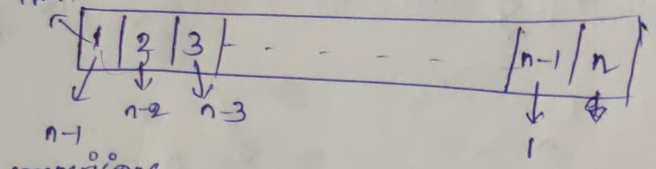
$a = b^k$ ⟨ $p > -1 \Rightarrow n^{\log_b^a} \log_b^{p+1} n$; $p = -1 \Rightarrow n^{\log_b^a} \log\log n$; $p < -1 \Rightarrow n^{\log_b^a}$

$a < b^k$ ⟨ $p \geq 0 \Rightarrow n^k \log^p n$; $p < 0 \Rightarrow n^k$

Bubble sort:

```
for(i=0; i<n-1; i++)
{
  for(j=0; j<n-i; j++)
  {
    if (A[j]>A[j+1])
    {
      temp=A[j];
      A[j]=A[j+1];
      A[j+1]=temp;
    }
  }
}
```

Comparisions in worst case:



$= (n-1) + (n-2) + \cdots + 2 + 1$

$= \frac{(n-1)(n)}{2} \Rightarrow O(n^2)$ → same in avg case