

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ T(n-1) + n - 1 & \text{if } n > 1. \end{cases}$$

Algorithm 1 Merge(A, p, q, r)

```
1   $n_1 = q - p + 1$ ;
2   $n_2 = r - q$ ;
3  Let  $L[1..n_1]$  and  $R[1..n_2]$  be new arrays
4  for  $i = 1$  to  $n_1$ 
5       $L[i] = A[p + i - 1]$ ;
6  for  $j = 1$  to  $n_2$ 
7       $R[j] = A[q + i]$ ;
8   $i = 1$ ;
9   $j = 1$ ;
10  $k = p$ ;
11 while  $i < n_1$  and  $j < n_2$ 
12     if  $L[i] \leq R[j]$ 
13          $A[k] = L[i]$ ;
14          $i++$ ;
15     else
16          $A[k] = R[j]$ ;
17          $j++$ ;
18      $k++$ ;
19 if  $j == n_2$ 
20     for  $m = k$  to  $r$ 
21          $A[m] = L[i]$ ;
22      $i++$ ;
```

Algorithm 2 BinarySearch($A, target$)

```
1   $low = 1$ ;  
2   $high = A.length$ ;  
3  while  $low \leq high$   
4       $mid = (low + high)/2$ ;  
5      if  $A[mid] == target$   
6          return  $mid$ ;  
7      if  $A[mid] > target$   
8           $high = mid - 1$ ;  
9      else  
10          $low = mid + 1$ ;  
11 return  $-1$ ;
```
