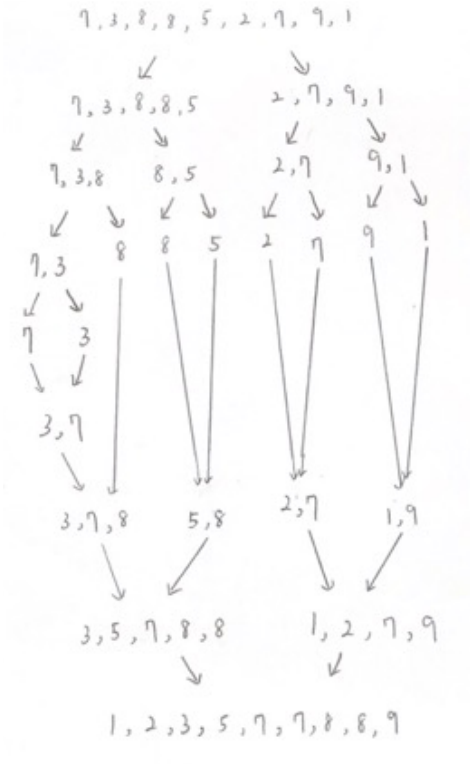
Q1: Suppose we have an array Arr = <7,3,8,8,5,2,7,9,1>. How to sort Arr using merge sort? Draw the sorting steps.

A1:



Q2:

Consider a sorted sequence of n numbers <a_1, a_2,...,a_n> stored in array A[1:N] and a value x. You need to design a **recursive binary search algorithm** to determine whether x is in the sequence or not. The algorithm should return TRUE if x is found and FALSE otherwise. The algorithm must have a worst-case time complexity of $O(log_2 n)$.

A2:

```
1: function BINARY-SEARCH(A, x, L, R)
2:     if L > R then
3:         return FALSE
4:     end if
5:     Mid = ⌊(L + R)/2⌋
6:     if A[Mid] == x then
7:         return TRUE
8:     else if A[Mid] > x then
9:         return BINARY-SEARCH(A, x, L, Mid − 1)
10:    else
11:        return BINARY-SEARCH(A, x, Mid + 1, R)
12:    end if
13: end function
```

The worst-case occurs when the target element x is not present in the array or is found at the last possible position. Since each round halves the size of the problem, the number of rounds required to reduce the problem size from n to 1 is $log_2 n$.