

# MERGE SORT: DIVIDE AND CONQUER

---

Link to practice handout

<https://bit.ly/Divide-and-Conquer-Practice>

# Divide and Conquer Algorithms

## Algorithm Approach:

- **Divide** a large problem into sub-problems
- **Solve** each sub-problem
- **Combine** the solutions of sub-problems to obtain the solution for the original problem

# Merge Sort Algorithm

**MergeSort(vector v)**

- Divide v into left half and right half
- Sort the left half, then sort the right half
- Combine (merge) the two sorted halves

Example run of  
mergesort

[ 7 2 ]

Example run of  
mergesort

[ 7 2 3 -1 ]

[ 7 2 5 3 -1 ]

Example run of  
mergesort

What is the height of the binary tree trace of mergeSort?

- A. 1.
- B. 2
- C. 3
- D. 4
- E. 5

Generalize the  
answer for an input  
vector of size n

## Running Time Analysis

[ 7      2      5      3      -1 ]

$T(n) = \# \text{ copy operations to split lists} +$   
 $\# \text{ comparisons to merge lists} +$   
 $\# \text{ function calls}$

[ 7      2      5 ]      [ 3      -1 ]

[ 7      2 ]      [ 5 ]      [ 3 ]      [ -1 ]

[ 7 ]      [ 2 ]

## Space Analysis

[7      2      5      3      -1]     $S(n) =$

[7      2      5]      [3      -1]

[7      2]      [5]      [3]      [-1]

[7]      [2]

How much additional space is used by the time  
mergesort reaches the base case?

- A.  $n * \log(n)$
- B.  $n + n/2 + n/4 + n/8 + \dots + 1$
- C.  $n + n/2 + n/4 + n/8 + \dots + 1 + \log(n)$
- D. Something else

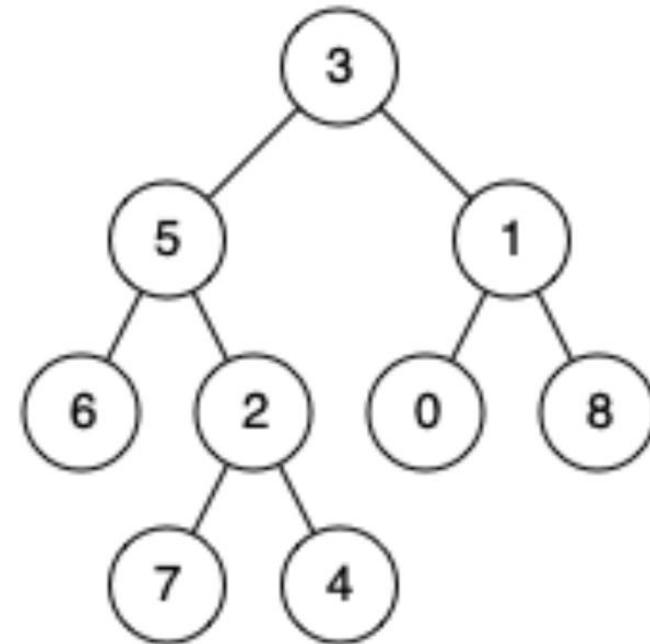
**Path:** a sequence of nodes in which each node is connected by an edge to the next.

**Ancestor(u):** any node that is on a path ending in u

**Descendant(v):** any node that is on a path starting from v

**Common ancestor(u, v):** any node that is the ancestor of both u and v

**Lowest Common ancestor(u, v):** deepest node in the tree that is a common ancestor of u and v



# Approach: Divide and Conquer

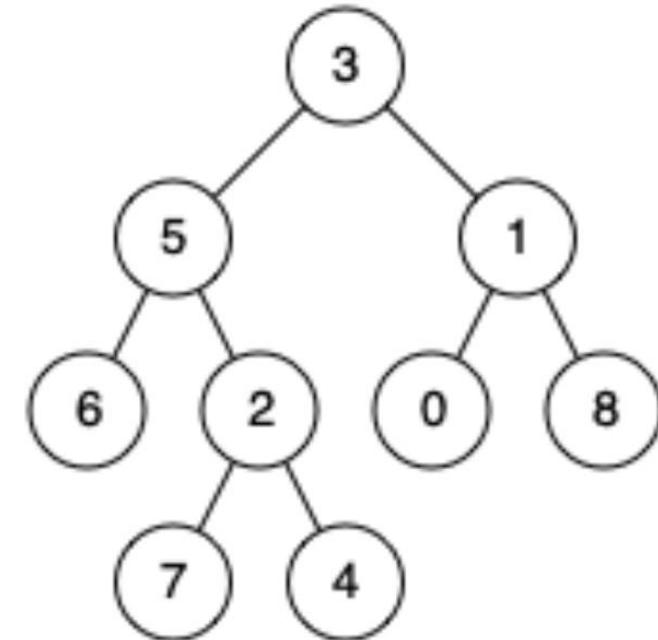
What is the LCA of each of the following?

5 and 1:

5 and 4:

6 and 7:

Discuss how you would solve the problem with your neighbor, trace your solution, describe in words, then implement Leetcode or handout



<https://bit.ly/Divide-and-Conquer-Practice>

<https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-tree/>