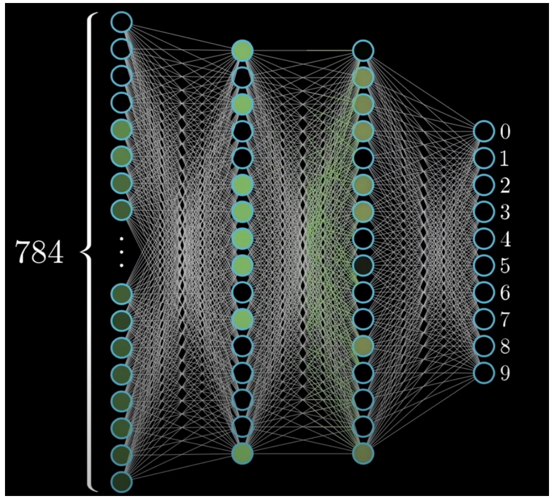
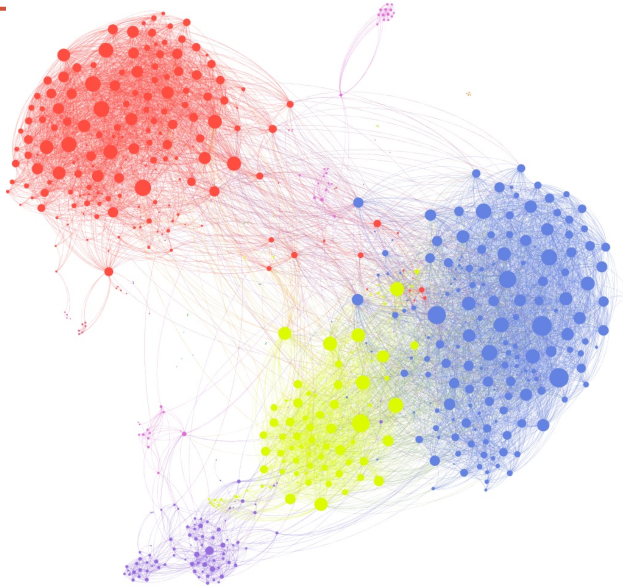
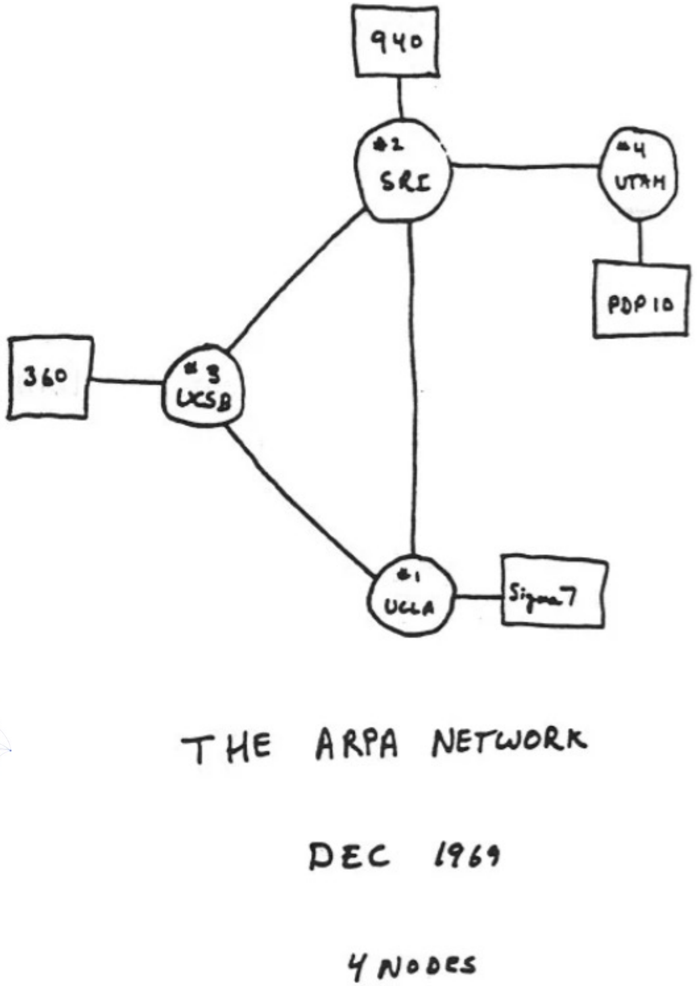
MERGE SORT


784
0
1
2
3
4
5
6
7
8
9




940
#2
SRI
#4
UTAH
PDP 10
360
#3
UCSB
#1
UCLA
Sigma7
THE ARPA NETWORK
DEC 1969
4 NODES

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
```

[7 2]

Example run of mergesort

[7 2 3 -1]

Example run of mergesort

[7 2 5 3 -1]

Example run of mergesort

What is the maximum depth 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

T(n) = # operations to split each list +
# comparisions to merge lists +
# function calls

```
[7    2    5    3    -1]

[7    2    5]   [3    -1]

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

[7]  [2]
```

Space Analysis

S(n) =

[7 2 5 3 -1]

[7 2 5] [3 -1]

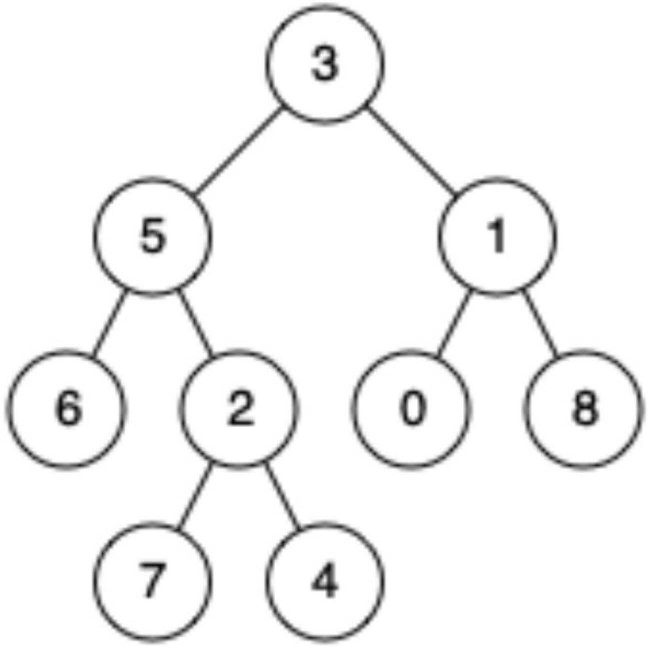
[7 2] [5] [3] [-1]

[7] [2]

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

A. n* log(n)
B. n + n/2 + n/4 + n/8 +….+ 1
C. n + n/2 + n/4 + n/8 +….+ 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

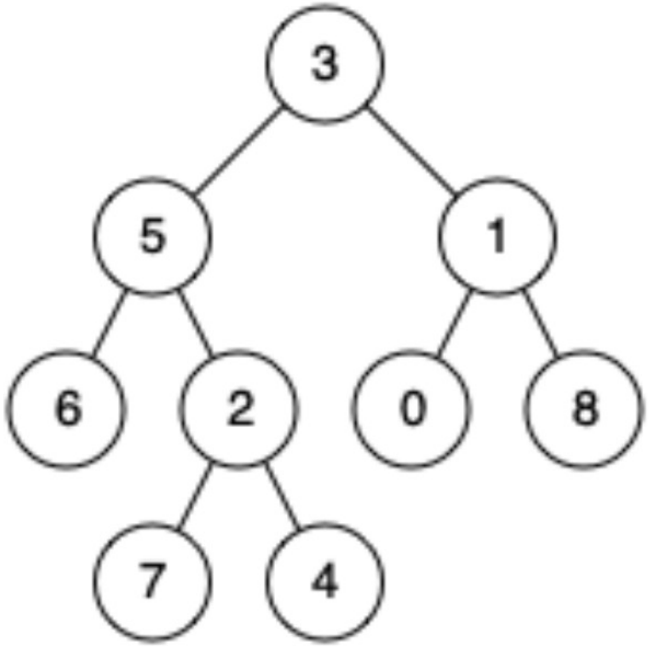
Descendant(v): any node that is on a path starting from v

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

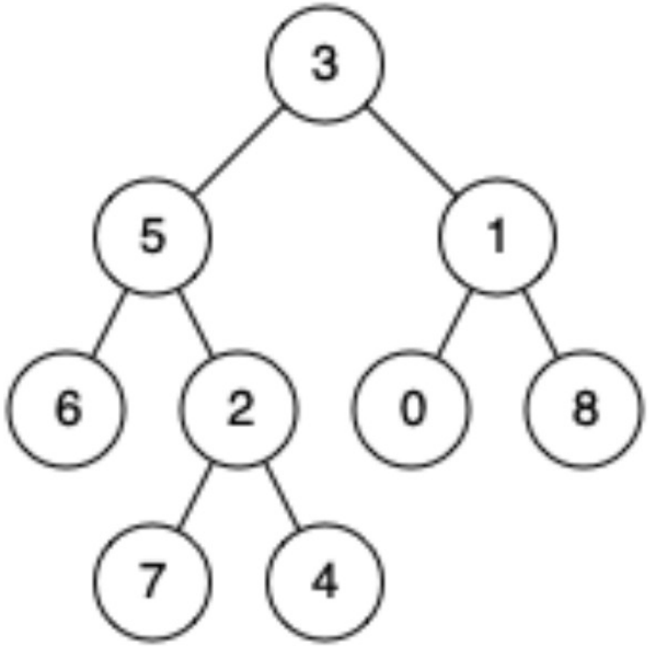
https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-tree/

Approach 1: Turn definitions into an algorithm



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

Approach 2: Divide and Conquer



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