

N-ary Tree

- 1) Given an N-ary tree where every node has at-most N children. How to serialize and deserialize it? Serialization is to store tree in a file so that it can be later restored. The structure of tree must be maintained. Deserialization is reading tree back from file.

Solution: Store an eof when you reach null

- 2) An N-ary tree has N sub-nodes for each node, it has M non-leaf nodes. Find the no of leaf nodes.

Solution: $1 + N + N^2 + \dots = M$

Red Black Trees

1) You have a weighted tree, consisting of n vertices. Each vertex is either painted black or is painted red. A red and black tree is called *beautiful*, if for any its vertex we can find a black vertex at distance at most x . The distance between two nodes is the shortest path between them. You have a red and black tree. Your task is to make it beautiful in the minimum number of color swap operations. In one color swap operation, you can choose two vertices of different colors and paint each of them the other color. In other words, if you choose a red vertex p and a black vertex q , then in one operation you are allowed to paint p black and paint q red. Find the minimum number of required actions.

Heap

- 1) Given two max heaps of size n each, what is the minimum possible time complexity to make a one max-heap of size from elements of two max heaps?

Ans: $O(n)$, explain how

- 2) Write an efficient program for printing k largest elements in an array. Elements in array can be in any order. (Repeated from last week, but heap wasn't taught till then)

Ans:

- 1) Build a Max Heap tree in $O(n)$

- 2) Use Extract Max k times to get k maximum elements from the Max Heap $O(k \log n)$

Time complexity: $O(n + k \log n)$

Explain other methods as well:

Method 1 (Use Bubble k times)

Method 2 (Use temporary array)

Method 3 (Use Sorting)

- 3) Sort a nearly sorted (or K sorted) array

Given an array of n elements, where each element is at most k away from its target position, devise an algorithm that sorts in $O(n \log k)$ time.

For example, let us consider k is 2, an element at index 7 in the sorted array, can be at indexes 5, 6, 7, 8, 9 in the given array.

Ans: Make a heap of k elements and then keep adding next element and deleting min element one at a time

$O(k) + O((n-k) \log K)$

Q1.

Find the largest multiple of 3

Given an array of non-negative integers. Find the largest multiple of 3 that can be formed from array elements.

For example, if the input array is {8, 1, 9}, the output should be "9 8 1", and if the input array is {8, 1, 7, 6, 0}, output should be "8 7 6 0".

Solution: Use 3 queues to separate numbers based on $n\%3$.

Q2.

Print the first n binary numbers using queue.

Solution: pop from queue, print it, insert it appended with 0 (and 1) back into queue

Q3.

Given a binary matrix, find out the maximum size square sub-matrix with all 1s.

Q4.

Given a singly linked list, check if it forms a cycle.

Solution: Use two pointers, increase one by 1, other by 2, and keep checking if they coincide.

Q5.

There are two singly linked lists in a system. By some programming error the end node of one of the linked list got linked into the second list, forming an inverted Y-shaped list. Write a program to get the point where two linked list merge.

Solution:

-> Use two for loops on pointers

-> Mark nodes as visited

-> Take difference between lengths of the two lists. Traverse forward on the longer list by the difference of lengths. Thereafter move parallelly on the two lists checking for coincidence.