Problem Set 5

Data Structures and Algorithms, Fall 2022

Due: October 13, in class.

Problem 1

Devise an algorithm to compute the median of an array $A[1, \cdots, 5]$ of distinct numbers using at most 6 comparisons. Instead of writing pseudocode, describe your algorithm using a *decision tree*: A binary tree where each internal node contains a comparison of the form "$A[i] \leq A[j]$?" and each leaf contains an index into the array.

Problem 2

The problem of merging two sorted lists arises frequently. We have seen a procedure for it as the subroutine MERGE in MERGESORT. In this problem, we will prove a lower bound of $2n - 1$ on the worst-case number of comparisons required to merge two sorted lists, each containing n items. First we will show a lower bound of $2n - o(n)$ comparisons by using a decision tree.

(a) Given $2n$ numbers, compute the number of possible ways to divide them into two sorted lists, each with n numbers.

(b) Using a decision tree and your answer to part (a), show that any algorithm that correctly merges two sorted lists must perform at least $2n - o(n)$ comparisons.

Now we will show a slightly tighter $2n - 1$ bound.

(c) Show that if two elements are consecutive in the sorted order and from different lists, then they must be compared.

(d) Use your answer to the previous part to show a lower bound of $2n - 1$ comparisons for merging two sorted lists.

Problem 3

(a) Develop an algorithm that, given n integers in the range 0 to k, preprocesses its input and then answers any query about how many of the n integers fall into a range $[a, b]$ in $O(1)$ time. Here, $a \in \mathbb{N}$ and $a \in \mathbb{N}$. Your algorithm should use $O(n + k)$ time for preprocessing.

(b) You are given an array of integers, where different integers may have different numbers of digits, but the total number of digits over all the integers is n. Develop an algorithm to sort the array in $O(n)$ time.

Problem 4

(a) In the algorithm QUICKSELECT we introduced in class, the input elements are divided into groups of 5. Will the algorithm work in linear time if they are divided into groups of 7? What about groups of 3? You need to justify your answer.

(b) Devise an algorithm to determine in $O(n)$ time whether an arbitrary array $A[1, \cdots, n]$ contains more than $n/4$ copies of any value.

Problem 5

In this problem, a *subtree* of a binary tree means any connected subgraph. In this problem, we say a binary tree is *complete* if every internal node has two children, and every leaf has exactly the same depth. Describe and analyze an algorithm to compute the largest complete subtree of a given binary tree. Your algorithm should return both the root and the depth of this subtree. See following for an example.

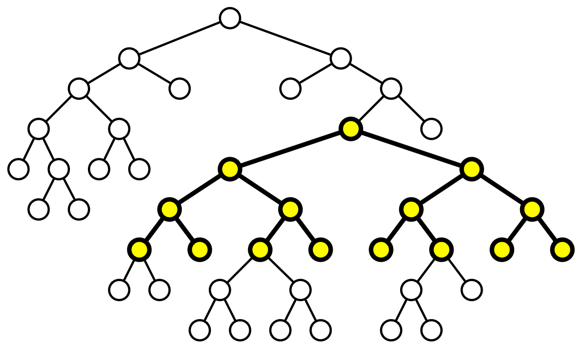

Figure 1: The largest complete subtree of this binary tree has depth 3.

Problem 6 [OJ Problem]

(Solve this problem on the OJ platform, do not hand in written solutions!)

The pre/in/post-order numbering of a binary tree labels the nodes of a binary tree with the integers $0, 1, \cdots, n-1$ in the order that they are encountered by a pre/in/post-order traversal. (See following for an example.) Suppose you are given a set of nodes with pre-order and in-order numbers assigned, then it can be shown that there is at most one binary tree with this pre/in-order numbering. Develop an algorithm to construct that tree.

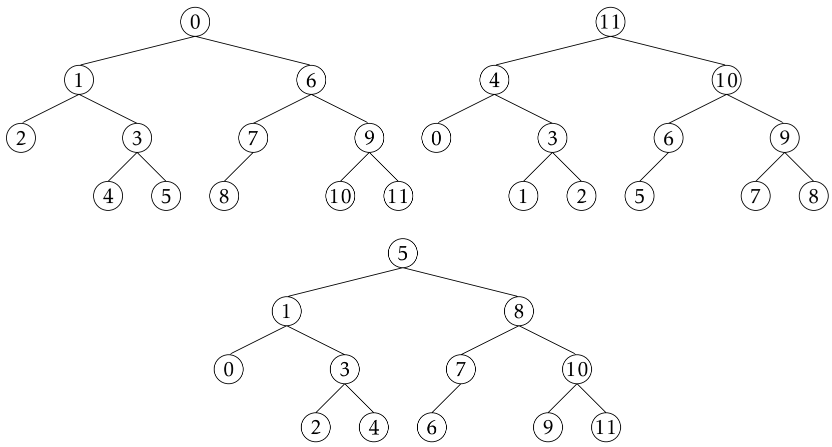

Figure 2: Pre-order, post-order, and in-order numberings of a binary tree.