

CS 473: Fundamental Algorithms, Fall 2014

Discussion 4

September 23, 2014

1. RECURRENCES Separators in trees.

Divide and conquer is a basic strategy and when we wish to apply it to graphs we need some additional properties about how the pieces of the graph interact. This is not always possible for all graphs but trees are simple. To this end suppose we have a tree $T = (V, E)$ on n nodes. A node v is said to be a balanced *separator* if removing v from T results in a forest in which no tree of the forest has more than $2n/3$ nodes. Prove that every tree T has a balanced separator.

Hint: Root T and consider the root and if it does not work pick an appropriate child etc. Describe a linear time algorithm to find such a separator. This can be generalized to the case where nodes have weights and the balance is with respect to the weights on the nodes. Can you formulate such a theorem?

Some times we wish to remove an edge of the tree. Say an edge e of T is a balanced separator if removing e results in two trees with each having at most $2n/3$ nodes. Give an example of a tree without such a separator. Suppose all nodes in T have degree at most $d + 1$. Show that there is an edge e such that $T - e$ has no component with more than $(1 - 1/d)n$ nodes.

2. LARGEST COMPLETE SUBTREE.

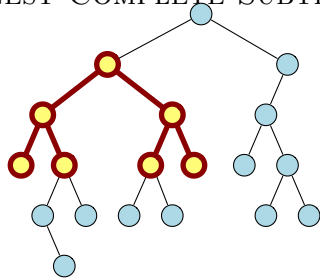


Figure 1: The largest complete subtree of this binary tree has depth 2.

A *subtree* of a binary tree is a connected subgraph, see Figure 1. A binary tree is *complete* if every internal node has two children, and every leaf has exactly the same depth. Describe and analyze a recursive algorithm to compute the *largest complete subtree* of a given binary tree. Your algorithm should return the root and the depth of this subtree.

3. ROD CUTTING.

Suppose we are given a steel rod of length n . Also, assume we are given an array $p[1 \dots n]$, where $p[i]$ is the price a rod of length i sells for. Given that we can make cuts for free (and that we only cut integer lengths), provide an algorithm that efficiently computes the maximum amount we can make by cutting up and selling our rod of length n .

4. k PARTITION.

Given a set of $S = \{\alpha_1, \dots, \alpha_n\}$ of n positive integers in the range 1 to m , decide, given a parameter k , if the numbers in S can be partitioned into k sets S_1, \dots, S_k , such that, for all i , $w(S_i) = w(S)/k$, where $w(X) = \sum_{x \in X} x$. What is the running time of your algorithm as a function of n, m and k .