Gray Houston

ghousto

0026483532

Project 4 Analysis Questions

**For all questions, the variable ‘*n*’ refers to the number of vertices and the variable ‘*m*’ refers to the number of edges.**

1. The asymptotic performance of my recursive tree height function is O(*n*), where *n* is the number of nodes in the tree. The algorithm basically works by finding the maximum height of the current children and returning that height plus 1, while watching for the leaf node corner case.

Because of the tree structure, each node can only be accessed by its parent. Therefore, once a parent node has recursively calculated the height of a child (and the algorithm starts moving back down the recursive stack), that child’s height will never be used again and that child node will not ever be accessed again. Proof for this statement:

Because the child’s height is only used to calculate the parent’s height, a child’s height is only needed exactly once. Therefore, once a parent’s height has been determined, it never needs to be accessed again. Furthermore, because the recursive accessing is essentially a DFS traversal of the tree, each node will be accessed once.

For this reason, throughout all of the recursive calls, each node will be accessed exactly once. Said another way, there will be *n* node accesses in the algorithm. Therefore, the asymptotic performance of the algorithm will be O(*n*), where *n* is the number of nodes in the tree.

It should be noted that there will be some overhead because of the recursion. However, as *n* increases and approaches its asymptotic boundary (i.e. as *n* approaches ∞), this recursive overhead becomes insignificant and negligible. Therefore, this overhead shouldn’t be considered for the asymptotic performance.

2.