Eric Bronner, Aedan Dispenza, Jason Davis, Timothy Yong
CS513 - Dr. Farach-Colton

Homework 3

Problem 1

a) Consider the subset $\alpha \subset A$ where A is all the possible subtrees of G, where G is the complete binary tree.

$$\forall a \in \alpha$$
$$\forall b_i \in a,\ b_i \text{ are leaves of } G$$
We can therefore say that a is some combination of the leaves in G
The total possible combinations of just the leaves of the tree are $(\frac{n+1}{2})!$
$$O(n!) \notin P$$

b) T_n can be a linked list.

Problem 2

Given a complete graph G s.t. each vertex represents an element in the graph, go through the entire graph, and weight the edges as the difference between each of the verticies.

Starting at the minimum vertex, follow the min path.

If a Hamiltonian Path exists from following the min path, then the path is sorted.

This algorithm is $O(n^2)$

Problem 3

Without an algorithm $HamP \in P$, this algorithm would need to find every possible spanning tree and therefore $BDST \in NP$, as you would use HamP to find each spanning tree

Problem 4

BuildCartesian(Tree T):

1. $DFS(T)$ to get the set of all edges E
2. $mergesort(E)$
3. remove the minimum edge and make it the root of the subtree
4. $L_subtree \leftarrow UnionFind(T_L)$
5. $BuildCartesian(T_L)$
6. $R_subtree \leftarrow UnionFind(T_R)$
7. $BuildCartesian(T_R)$

Analysis:

DFS runs in $O(|E|)$, Mergesort runs in $O(|E|log|E|$, removing the minimum subtree is $O(1)$, and the UnionFinds together run in $O(|V|log|V|)$

Thus, the sum of all recursive calls is $O(|V|log|V|)$ by the Master's Theorem