Problem Set 13

Due Date .... December 13, 2022
Name .... **Alex Ojemann**
Student ID **109722375**
Collaborators .... **None**

Contents

1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. (See this short intro to LaTeX plus other resources on Canvas.)
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You are welcome and encouraged to collaborate with your classmates, as well as consult outside resources. You must **cite your sources in this document. Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.
- Posting to **any** service including, but not limited to Chegg, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.
- You **must** virtually sign the Honor Code (see Section 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

- My submission is in my own words and reflects my understanding of the material.
- Any collaborations and external sources have been clearly cited in this document.
- I have not posted to external services including, but not limited to Chegg, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

(I agree to the above, Alex Ojemann). □

3 Standard 24: Complexity Classification

For the following problem, figure out whether it is in P or NP-complete, and prove it. For P this means giving a poly-time algorithm. For NP-complete, this means three things: (1) showing it's in NP, (2) giving a reduction from an NP-complete problem we cover in class, and (3) showing that reduction runs in polynomial time. If you give a reduction, **prove** that your reduction is correct.

(a) Let T_1, T_2 be rooted binary trees, with specified root nodes r_1, r_2. We say that T_1 and T_2 are *isomorphic* if there exists a bijection $\varphi : V(T_1) \to V(T_2)$ such that $\varphi(r_1) = r_2$, and for all $u, v \in T_1$, we have that:

$$\{u, v\} \in E(T_1) \iff \{\varphi(u), \varphi(v)\} \in E(T_2).$$

Given two rooted binary trees T_1 and T_2, we are interested in the algorithmic problem of deciding whether T_1 and T_2 are isomorphic.

Proof. This problem is in P.

A polynomial time algorithm to solve this problem is as follows:

isIso(node1,node2)

if node1 is null and node2 is null return true

if node1 is null and node2 isn't return false

if node2 is null and node1 isn't return false

if the contents of node1 don't equal the contents of node2 return false

else return (isIso(left child of node 1, left child of node 2) and isIso(right child of node 1, right child of node 2) or (isIso(left child of node 1, right child of node 2) and isIso(right child of node 1, left child of node 2)

The initial call of this function is isIso(root of T_1,root of T_2)

The complexity of this algorithm is O(n) where n is the minimum of the number of nodes in tree 1 and the number of nodes in tree 2 because it checks each node in both trees to see that they are all isomorphic and if the trees have different numbers of nodes it will end when the first null node appears in one tree but doesn't have a corresponding null node in the other tree with the same parent.

Since O(n) is polynomial time, this problem must be in P. □