Name: Jaryd Meek
ID:
Collaborators: Noah Nguyen, Emily Parker

CSCI 3104, Algorithms **Due THURSDAY, APRIL 29, 2021**
Problem Set 10 (50 points) **Spring 2021, CU-Boulder**

Advice 1: For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.
Advice 2: Verbal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.

Instructions for submitting your solution:

- The solutions **should be typed** and we cannot accept hand-written solutions. Here's a short intro to Latex.
- You should submit your work through **Gradescope** only.
- The easiest way to access Gradescope is through our Canvas page. There is a Gradescope button in the left menu.
- Gradescope will only accept **.pdf** files.
- It is vital that you match each problem part with your work. Skip to 1:40 to just see the matching info.

Name: Jaryd Meek
ID:
Collaborators: Noah Nguyen, Emily Parker

CSCI 3104, Algorithms **Due THURSDAY, APRIL 29, 2021**
Problem Set 10 (50 points) **Spring 2021, CU-Boulder**

1. Let P_1, P_2 be two problems such that $P_1 \leq_p P_2$. That is, we have a polynomial-time reduction $r : P_1 \to P_2$. If we assume $P_2 \in P$, explain why this implies that $P_1 \in P$.

Solution:

- $P_1 \leq_p P_2$
- $r : P_1 \to P_2$
- $P_2 \in P$

We know that P_1 is as difficult as P_2 and vice-versa. Therefore, an algorithm used to solve P_2 can be used to solve P_1. We are able to reduce P_1 to P_2 if there exists a function that takes any input x for P_1 and transforms it to an input $f(x)$ of P_2, such that the solutions to P_2 on $f(x)$ is the solution to P_1 on x. Therefore by the defintion of reduction, $P_1 \in P$.

Name: Jaryd Meek
ID:
Collaborators: Noah Nguyen, Emily Parker

2. Recall the k-Colorability problem.

   - Input: Let G be a simple, undirected graph.
   - Decision: Can we color the vertices of G using exactly k colors, such that whenever u and v are adjacent vertices, u and v receive different colors?

   It is well known that k-Colorability belongs to NP, and 3-Colorability is NP-complete. Our goal is to show that the 4-Colorability problem is also NP-complete. To do so, we reduce the 3-Colorability problem to the 4-Colorability problem. We provide the reduction below. The questions following the reduction will ask you to verify its correctness.

   Reduction: Let G be a simple, undirected graph. We construct a new simple, undirected graph H by starting with a copy of G. We then add a new vertex t to H, and for each vertex $v \in V(G)$ we add the edge tv to $E(H)$.

   Your job is to verify the correctness of the reduction by completing parts (a)–(c) below, which completes the proof that 4-Colorability is NP-complete. For the rest of this problem, let G be a graph, and let H be the result of applying the reduction to G.

   (a) Suppose that G is colorable using exactly 3 colors. Argue that H is colorable using exactly 4 colors.

   (b) Suppose that H is colorable using exactly 4 colors. Argue that G is colorable using exactly 3 colors.

   (c) Let n be the number of vertices in G. Carefully explain why H can be constructed in time polynomial in n. [**Hint:** Count the number of vertices and edges we add to G in order to obtain H.]

   Solution:

   (a) If H has one more vertex than G, and G is colorable by 3 colors, all the nodes that are in G can color to the corresponding node in H. This means that the additional node in H can be colored with a 4th color. This is true since H only has one additional node that can have it's own color. This separate node (t), will also have a separate color because it is adjacent to all other nodes.

   (b) If H is colorable with 4 colors, this means that the node t has a different color because it is adjacent to all other nodes. Then the remaining parts of H without the node t is colored with 3 colors. The remainder of the graph is the same as G, therefore, removing node t leaves graph G that is colored with 3 colors.

   (c) To create H from G, we add 1 node then add n edges between that node. The total time will therefore be $\Omega(n+1) = \Omega(n)$ and will have to traverse through all nodes in G one time.