Fall 2024 COT 4210 Exam #4
November 21, 2024
Sheet 1: Class P

Last Name: _____ , First Name: _____

1) (15 pts) Prove that the language, L, below, belongs to the class P. (Note: You may assume that the grader fully understands what a depth first search or breadth first search does, but you must state its run time properly.)

L = { <G> | G is an undirected, unweighted graph, which, with the addition of one edge, can be connected.}

2) (10 pts) Prove that the language, L, below, belongs to the class P.

L = { <S> | S is a list where no two integers in the list are the same.}

Fall 2024 COT 4210 Exam #4
November 21, 2024
Sheet 2: Class NP

Last Name: _____ , First Name: _____

3) (15 pts) Let BIGDIVISOR = { (n, k) | n has at least one proper divisor greater than k. }.

(a) Prove that BIGDIVISOR is in NP.

(b) Explain why regular trial division (dividing n by each integer in between k+1 and n-1, inclusive), doesn't prove that BIGDIVISOR is in P.

4) (20 pts) For this question, write a verifier for 3-SAT. The verifier will take in an integer array, **phi**, (several arrays of size 3), where each array represents a clause as well as a String, **s**, representing the truth setting of each variable to verify. Let the variables be x_1, x_2, …, x_k. A positive integer i represents the variable x_i and the negative integer -i represents the variable $\overline{x_i}$. Thus, the clause $(x_2 \vee \overline{x_5} \vee \overline{x_6})$ would be represented by the array [2, -5, -6]. The string s will store the truth setting of each variable to verify. If the character at index i is 'T', then variable x_{i+1} is set to true and if the character at index i is 'F', then the variable x_{i+1} is set to false. Your method should return true if the truth setting in **s** satisfies the boolean formula represented in the 2D array **phi**, and false otherwise. (Note: n and k are NOT valid variable names inside of the code unless you define them…) You may assume the input is well-formed; that every variable described in phi is represented in s. (So, you don't need to worry about array out of bounds errors if you write your code in a logically correct manner.)

```
import java.util.*;

public class e4q4 {

     public static boolean verify(int[][] phi, String s) {









     }
}
```

Fall 2024 COT 4210 Exam #4
November 21, 2024
Sheet 3: Class NP-Complete

Last Name: _____ , First Name: _____

5) (15 pts) In homework 6, we dealt with the problem 4-SAT, but never proved that it's NP-Complete. Prove that 4-SAT is NP-Complete. Do all parts of the proof.

6) (20 pts) A cut in a graph is the partition of a graph's vertices into two disjoint sets, *S* and *T*. The weight of a cut is defined as the sum of edge weights of the edges with one endpoint in *S* and the other endpoint in *T*. For the purposes of this problem, the graph is undirected. Let MAX-CUT = { <G, k> | G has a cut of weight k or more}. It can be proven that this problem is NP-Complete, thus for this problem, assume that it's known that this language is an NP-Complete language.

Consider a language called MIN-GROUP-PENALTY where the input is a set of n people who are to be placed on two disjoint teams at a cost of k or less. For each pair of distinct people, a and b, the cost of putting those two people on the same group is cost(a, b). (Note that cost(a, b) = cost(b, a). The theory here is that maybe a and b don't like each other, so we'll have issues if we place them on the same team and this is encapsulated in the cost function.) We can specify the language as follows:

MIN-GROUP-PENALTY = { <C, k> | C stores the cost matrix for all pairs of people where there exists a way to place all people onto two disjoint teams at a total cost of k or less.}

(a) Prove that MIN-GROUP-PENALTY is in NP.
(b) Via reduction from MAX-CUT, prove that MIN-GROUP-PENALTY is NP-Complete.

7) (5 pts) In what city is the Indianapolis Motor Speedway? _____