# Problem Set 11

Due Date ..... December 1, 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. You do **not** have to prove that your reduction is correct.

(a) A city is having issues with its $n \times m$ grid of street lights: each light is of type $A$, $B$, or $C$, but due to electrical problems, the city must shut off all lights of type $C$. Additionally, for each column of the grid, the city must shut off all lights of type $A$ in that column or all lights of type $B$ in that column. Given the type of each light in the grid, the city would like to decide whether it is possible to resolve the electrical problems while still leaving one light on in every row.

*Proof.* This problem is in NP because any candidate solution is verifiable by checking each row and determining whether at least one light is on for all rows and this takes O(n*m) time.
We can reduce SAT to this problem. Each boolean in SAT can be represented by a street light in this problem, which would be true if the street light is on or false if it's off. The boolean expression would be of the form $(L11 \vee L12 \vee ... \vee L1m) \wedge (L21 \vee L22 \vee ... \vee L2m) \wedge ... \wedge (Ln1 \vee Ln2 \vee ... \vee Lnm)$ where $Lab$ is the boolean representing the light at row a column b. This is because we are checking whether any light in all the rows are working, so we or together each light in a given row then and these blocs for all the rows. All lights of type C correspond to booleans already set to false and the boolean expression will have the same constraint that for each column of the grid, the city must shut off all lights of type A in that column or all lights of type B in that column. In order to enforce this constraint, each boolean will also have a type A, B, or C which correspond to the type A, B, and C lights.
Any satisfactory solution of this version of SAT will also be a satisfactory solution of the city lights problem, and any unsatisfactory solution of this version of SAT will also be an unsatisfactory solution of the city lights problem. Thus if this version of SAT has a satisfactory solution, so will the city lights problem.
This reduction runs in no greater than polynomial time because representing each boolean as a light takes O(n*m) time and representing the type A, B, or C booleans as type A, B, or C lights takes O(n*m) time. Therefore, the described city lights problem is NP-complete. □