CS 5150/6150 Graduate (Advanced) Algorithms
Fall 2022
Assignment 7: Computational Complexity
Total Points: 40

Submission notes

- Due at 11:59 pm on **Tuesday, December 6, 2022**.
- Solutions must be typeset (not hand-written or scanned).
- You should strive to write clear, concise solutions to each problem, ideally fitting on a page or less. The easier your work is to read & follow, the easier it is for the TAs to award you points.
- Upload a PDF version of your completed problem set to Gradescope.
- Teaching staff reserve the right to request original source/tex files during the grading process, so please retain these until an assignment has been returned.
- Please remember that for problem sets, *collaboration with other students must be limited to a high-level discussion of solution strategies.* If you do collaborate with other students in this way, you must identify the students and describe the nature of the collaboration at the top of your homework submission. **You are not allowed to create a group solution, and all work that you hand in must be written in your own words. Do not base your solution on any other written solution, regardless of the source.**

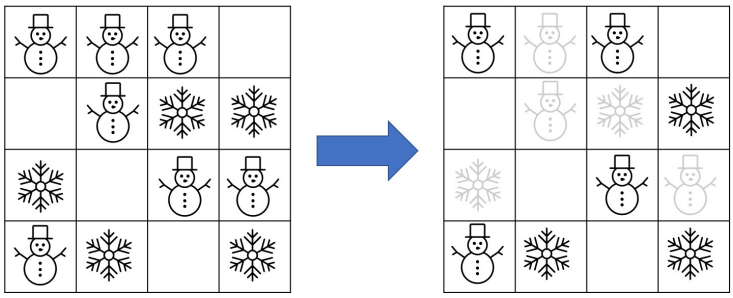
1. Consider the BoxDepth problem: given a set S of n axis aligned rectangles in the plane (that is, their sides are parallel to the x and y axes), is there a subset $X \subseteq S$ of size at least k such that all of the rectangles in X share a common point (that is, there are values x^* and y^* so that the point (x^*, y^*) is contained in every rectangle in X)? We assume that a rectangle contains all points on its boundary (in addition to those strictly inside).

   (a) **(10 points)** Describe a polynomial-time reduction from BoxDepth to Clique. You must include a description in English of how to transform an arbitrary instance of one problem into the other, clearly note the size of the new instance with respect to the size of the original instance, state and argue the runtime of your transformation. Further, be sure to include both directions of the correctness proof.

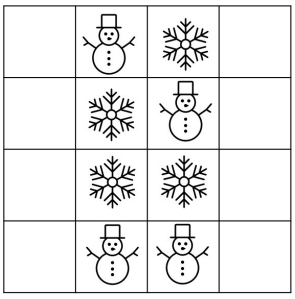
   (b) **(7 points)** Describe (in English) a polynomial-time algorithm for BoxDepth and analyze its runtime. [*Hint:* $O(n^3)$ *should be easy.*]

   (c) **(3 points)** Why don't these two results imply that P = NP?

2. **(20 points)** Many games that humans are quite good at are actually computationally hard to solve. For example, your little cousin brought over their new holiday puzzle game last weekend. When you press the Play button, the electronic puzzle displays an $n \times m$ grid of squares. Each square is either empty, occupied by a snowman, or occupied by a snowflake. The goal of the game is to remove some of the snowmen/snowflakes (by clicking on them) so that those remaining on the board satisfy two conditions: (1) no row has all of its squares empty, and (2) no column contains both a snowman *and* a snowflake. If you can accomplish this, we call the puzzle instance *solvable*. An example of a solvable configuration and one of its (many) solutions is below.



However, for some initial configurations, satisfying these two conditions is impossible. An example of such an *unsolvable* instance is below.



Prove that it is NP-hard to determine whether a given initial configuration is solvable.
Hint: a puzzle is solvable if there the rules are ***satisfiable*** *by some subset of snowmen/snowflakes.*