Problem set 5

Date : Mar 25, 2022 Algorithms-S22

Instructions

- The problem sets are not for submission and will not be graded. However, you are strongly encouraged to spend some time thinking about the questions. This might be helpful in internalizing some of the things that we would discuss in the lectures.
- To get the most out of problem sets, you are strongly encouraged to think about the problems on your own before discussing with others, consulting any references or looking at hints (that some of the problems might have).

Problems

1. Suppose you have access to a subroutine that magically solves SAT in constant time, i.e, given a Boolean formula Φ as input, this subroutine correctly outputs in $O(1)$ time where Φ is satisfiable or not.

   Use this subroutine to design an efficient algorithm that takes as input a Boolean formula Ψ and outputs an assignment to the variables that satisfies Ψ if Ψ is satisfiable or correctly states that Ψ is unsatisfiable.

2. Consider the problem P: given a set of axis parallel rectangles in the plane, the goal is to determine the size of the largest subset of these rectangles that contain a common point.

   - Describe a polynomial time reduction from P to the maximum independent set problem.
   - Design an efficient algorithm for this problem.
   - Do these results together imply that $\mathsf{P} = \mathsf{NP}$? Why or why not ?

3. Design an efficient algorithm that when given as input a 3CNF formula where every clause has all true literals or all false literals decides whether the formula is satisfiable.

4. MaxSAT is the problem where given as input a 3CNF formula, the goal is to compute the maximum number of clauses that can be satisfied by any assignment to the variables.

   - Show that MaxSAT is NP-hard.
   - Show that for every 3CNF formula, there is an assignment to the variables that satisfies at least a $7/8$ fraction of all clauses.

5. For an undirected graph $G = (V, E)$, a dominating set in G is a subset S of vertices such that every vertex in G is either in S or adjacent to a vertex in S. The DominatingSet problem asks, given a graph G and an integer k as input, does G has a dominating set of size at most k.

   Prove that this problem is NP hard.