**CSE 102: Spring 2021**
**Advanced Homework 7**
Approximate Solutions to NP Problems
or
Polynomial Time Algorithms for Simplified NP Problems
(10+ points)

Follow Advanced Homework Citation Guidelines.
Attempt **ONE** of the following problems.

1. **2-colorable Graphs:** (2 points) Write an algorithm to determine if a graph is 2-colorable. The algorithm should run in $\theta(v+e)$ time, where $v$ is the number of vertices and $e$ is the number of edges, and produce a 2-coloring if one exists.

2. **2SAT:** (5 points) Design an algorithm to solve 2SAT in polynomial Time.

3. Prove that **0-1 knapsack problem** is excellent in approximable hierarchy. Design a polynomial time greedy algorithm that achieves $(1+\epsilon)$-approximation for any $\epsilon > 0$. In other words, there is a polynomial approximation that comes arbitrarily close to the maximum value.This is as good as it gets!

4. Prove that **traveling salesman problem** is worst in approximable hierarchy. In other words, assuming $P \neq NP$, there does not exist any polynomial time algorithm that can achieve a k-approximation for any constant $k \geq 1$. In other words, there is no polynomial approximation that can guarantee the optimal solution within any constant. [Hint: If such a constant exists, then one can solve Hamiltonian (Rudrata) Cycle Problem in polynomial time.] This is as bad as it gets!

5. Consider a **special case of 3SAT** in which all clauses have exactly three literals, and each variable appears exactly three times. Show that this problem can be solved in polynomial time. Recall a literal is either a variable or its negation. For example, $x$ is a variable, whereas $x$ and $\overline{x}$ are literals of the same variable. [Hint: Create a bipartitie graph with clauses on the left, variables on the right, and edges whenever a

variable appears in a clause. Use Exercise 7.30 of DPV to show that this graph has a matching.]

6. **Steiner Tree Problem**: 9.6 in DPV.

7. Problem 35.2.3 CLRS on *closest-point heuristic* for building an approximate **traveling salesman tour**.

8. Problem 35.2.4 CLRS on **bottleneck traveling salesman problem** where the most costly edge in the cycle is minimized.

9. Problem 35.4 CLRS on **Maximum Matching Problem**. A matching is a set of edges such that no two edges in the set are incident on the same vertex. A maximal matching is a matching that is not a proper subset of any other matching. A maximum matching is a matching of maximum cardinality, that is number of edges in a maximum matching is larger than any other matching. A maximal matching need not be a maximum matching.

10. Problem 3 in Chapter 11 of KT on **Largest Total Sum Subset Problem**.

11. Problem 7 in Chapter 11 of KT on **e-Commerce Advertising**.