Optional

General Directions: If you are asked to provide an algorithm, you should clearly define each step of the procedure, establish its correctness, and then analyze its overall running time. There is no need to write pseudo-code; an unambiguous description of your algorithm in plain text will suffice.

All the answers must be typed, preferably using LaTeX. If you are unfamiliar with LaTeX, you are strongly encouraged to learn it. However, answers typed in other text processing software and properly converted to a pdf file will also be accepted. Before submitting the pdf file, please make sure that it can be opened using any standard pdf reader (such as Acrobat Reader) and your entire answer is readable. **Handwritten answers or pdf files that cannot be opened will not be graded and will not receive any credit.**

Finally, please read the detailed collaboration policy given on the course website. You are **not** allowed to discuss homework problems in groups of more than 3 students. **Failure to adhere to these guidelines will be promptly reported to the relevant authority without exception.**

Problem 1
Give a divide and conquer algorithm to find the closest pair of points in a set of points on the two-dimensional plane.

Problem 2
Give an $O(n)$-time algorithm to check if an undirected graph contains a cycle.

Problem 3
Suppose you are given a directed graph with edge lengths that are positive integers. Give an algorithm to find the shortest path from vertex s to vertex t that uses an odd number of edges. Your algorithm should run in the same time as Dijkstra's algorithm.

Problem 4
Given a graph $G = (V, E)$ and an MST T on G, describe an $O(n)$ time algorithm to find a new MST if a new edge is added to the graph G.

Problem 5
Give an example where the max $s - t$ flow is less than the capacity of some $s - t$ cut.

Problem 6
Consider a random process where you are given a deck of n cards, and in each step, you select a card uniformly at random and put it back in the deck before the next draw. Define a random variable X to the number of draws after which you have drawn each card at least once. Find the expectation of X (in $\Theta(\cdot)$ notation).

Problem 7
Give an ILP formulation for the 0/1 knapsack problem and a fractional LP formulation for the fractional knapsack problem. Give the dual of the latter.

Problem 8
A graph coloring is a labeling (say, by color) of graph vertices such that no two adjacent vertices share the same label (color). The graph coloring problem is the problem of determining whether there is a graph coloring with k colors. Show that the graph coloring problem with $k = 3$ is NP-complete. (Give a polynomial time reduction of the 3-SAT problem to graph coloring with 3 colors. Then, show that graph coloring with 3 colors is in NP.)

Problem 9
Let $G = (V, E)$ be an undirected graph. In the max-cut problem, we need to find the cut containing the maximum number of edges. Consider the following GreedyCut algorithm: start with an arbitrary cut, and repeatedly move any vertex to the other side if that increases the size of the cut.

(a) Prove that the GreedyCut algorithm runs in polynomial time.

(b) Prove that GreedyCut is a 2-approximation algorithm for the max-cut problem.

Problem 10

A *maximal* matching is a matching such that no edge can be added to the matching edges while preserving the matching property.

(a) Give an example of a maximal matching that is *not* a maximum matching.

(b) Prove that any maximal matching is a 2-approximation of a maximum matching.

Problem 11

Suppose you are given an undirected graph G and you want to find the shortest path from vertex s to vertex t. Assume you are given an algorithm A that outputs shortest s-t path with probability $1/2$ and outputs a sub-optimal path with probability $1/2$. Additionally, assume A deterministically runs in time $O(t)$. Design the following two algorithms using A.

(a) Give a Monte Carlo algorithm that computes the shortest s-t path in G with an expected runtime of $O(t \log n)$. Recall that a Monte Carlo algorithm is correct with high probability (in other words, the algorithm is incorrect with probability $1/n^c$ for a constant c).

(b) Suppose someone gives you the value of the shortest path s-t path in G (but not the shortest path itself). Again using A, give a Las Vegas algorithm that returns the vertices in the shortest path and has an expected runtime of $O(t)$. Recall that a Las Vegas algorithm must always output the correct answer, but the running time need only hold in expectation.