CS 530: Advanced Algorithms

Homework 1: Review and Intro to Approximation Algorithms

Fall 2025

1 Problem 1: TSP (10 points)

(a) Show that metric TSP is equivalent to the following problem: given a weighted graph G, find the cheapest walk that visits every vertex at least once and returns to the starting point.

(b) In the metric path TSP problem, we are given special vertices s and t and need to compute the cheapest Hamiltonian path starting at s and ending at t. Give a 2 approximation for this problem.

2 Problem 2: Knapsack (15 points)

(a) Modify the algorithm we went over in class so that it runs in time $O(n \cdot \min(W, V))$, where W is the capacity of the knapsack and V is the value of the optimal solution. **Hint:** What solutions do you really need to keep around in each subproblem?

(b) Modify the algorithm so that it also returns an optimal solution, not just its value.

(c) Show how to modify the FPTAS to improve the running time to $O(n^2/\epsilon)$. **Hint:** Start by running the greedy 2 approximation for knapsack.

3 Problem 3: Implementation of Knapsack (10 points)

Implement the dynamic program for knapsack you obtained in 2(a) and run it on the instance posted here for bounds $W = 10000000$ and $W = 20000000$. Turn in your code and write the answers you got here.

4 Problem 4: k-suppliers (10 points)

In the k-supplier problem, the input is a positive integer k and a set of vertices V equipped with a metric $c : V \times V \to \mathbb{R}_{\geq 0}$. The vertices is partitioned into a set of potential *suppliers* $R \subseteq V$ and a set of *customers* $C = V \setminus R$. The goal is now to find the set S of k suppliers so as to minimize $\max_{i \in C} d(i, S)$ where $d(i, S)$ is the distance from customer i to its closest supplier in S.

In other words, we want to open k suppliers so as to minimize the *maximum* distance any customer has to travel to a supplier. Give a 3 approximation for this problem.