

CSE 551: Homework 5 (50 points)

November 2018

Submission Instructions: Deadline is **11:59pm on 11/25**. Late submissions will be penalized, therefore please *ensure* that you submit (file upload is completed) before the deadline. Submit your answers in a single PDF file. As usual, we will grade *any two* problems. Hence, please answer all the questions.

1. Input: Points $X = \{x_1, x_2, \dots, x_n\}$ with underlying distance metric $d(.,.)$ and an integer k .

Output: A partition of the points into k clusters C_1, C_2, \dots, C_k .

Goal: Minimize the diameter of the clusters,

$$\max_j \max_{x_a, x_b \in C_j} d(x_a, x_b)$$

One approach to solve the problem is as follows:

- Pick any point $\mu_1 \in X$ as the first cluster center
- for $i = 2$ to k :
 - Let μ_i be the point in X farthest from $\mu_1, \mu_2, \dots, \mu_{i-1}$ (i.e.) that maximizes $\min_{j < i} d(., \mu_j)$
- Create k clusters: $C_i =$ all $x \in X$ whose closest center is μ_i

Will this approach give the optimal solution? If yes, please justify. If no, then provide a counter example, where such approach will fail and an *upper bound on the error (displacement from the optimal)*.

2. Consider an Activity Selection Problem (ASP). There are rewards $r_i \in R$ associated with each activity A_i . An activity earns its reward, if it can be scheduled without violation, with the other activities in the schedule. Develop an algorithm to determine the set of activities which maximizes the reward.

3. Let the following matrix denote the distance between 5 cities,

$$G = \begin{bmatrix} 0 & 2 & 3 & 4 & 5 \\ 3 & 0 & 10 & 4 & 3 \\ 6 & 10 & 0 & 9 & 2 \\ 8 & 5 & 6 & 0 & 10 \\ 2 & 3 & 9 & 6 & 0 \end{bmatrix}$$

Using a dynamic programming technique, find the optimal Traveling Salesman tour from the problem instance given above. Show all your work.

4. Given n points in a 2D plane, the objective is to construct a tour passing through every point ONCE. You may consider Euclidean distance. Consider the following algorithm.

Compute the minimum spanning tree of the instance mentioned above. Given a spanning tree, one can convert it into a traveling salesman tour as follows. A reasonable short tour can be computed by traversing twice around the tree. Given a minimum spanning tree T we can start at any vertex s and take a path based on the depth-first search on the tree from s . In particular whenever we visit a new vertex v from vertex u we traverse the edge from u to v and when we are done visiting everything reachable from v we then back up on this same edge, traversing it from v to u . This way every edge in our path is traversed exactly twice, and we end the path at our initial vertex. If we view each undirected edge as two directed edges, then this path is a so-called Euler tour of the tree—i.e. a cycle in a graph that visits every edge exactly once. Since T spans the graph, the Euler tour will visit every vertex at least once, but possibly multiple times. This is better explained with the following example,

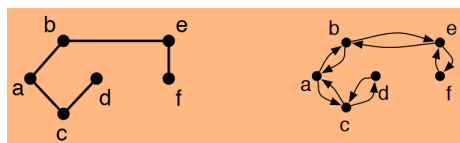


Figure 1: Twice Around MST



Figure 2: Shortcut Edges

Since it is possible to take an edge from any vertex to any other, we can take shortcuts to avoid visiting vertices multiple times. More precisely what we

can do is when about to go back to a vertex that the tour has already visited, instead find the next vertex in the tour that has not been visited and go directly to it. We call this a shortcut edge.

By the triangle inequality the shortcut edges are no longer than the paths that they replace. Thus by taking shortcuts, the total distance is not increased.

Will this approach give the optimal solution? If yes, please justify. Else, give a counter example where this fails. Also, provide an upper bound on the error (difference between the optimal and the solution obtained from this method) and explain your reasoning.