CSE 417 25au Homework 4: Graph Algorithms

Released: Friday, October 17, 2025 @ 11:30am

First due by: Friday, October 24, 2025 @ 11:59pm

Last resubmissions by: Wednesday, November 5, 2025 @ 11:59pm

Instructions

For Problems 7 and 8, you have four options for submission:

- **Film a video in which you explain your solution.** See the Homework Guide for more details.
- **Use LaTeX to type your solutions.** A template is provided in the "Tasks" page of the course website, if you like.
- **Use Google Docs or Microsoft Word to type your solutions.** If doing so, please use the Equation Editor to ensure that any equations are legible and easy to read.
- **Handwrite your solutions on paper or digitally.** Please write neatly and if on paper, scan in black/white mode, not grayscale.

We prefer either video or LaTeX, but accept any of the 4 options.

A few more reminders:

- **Submit all problems on Canvas.** Each problem should have its own submission. *Do not* submit one large file containing answers to several problems.
- **Suggested word counts are rough guidelines.** We won't actually count, but if your writing is verbose to the point of obscuring your main argument, we may ask you resubmit more concisely.
- **Review the collaboration policy in the syllabus**. Collaboration is encouraged but strict rules apply, and remember to cite your collaborators.
- If you don't finish in time, we encourage you to be honest and just upload what you have so far. Resubmission won't cost you anything, and we can give you timely feedback on your partial progress by submitting on time.

Happy problem solving!

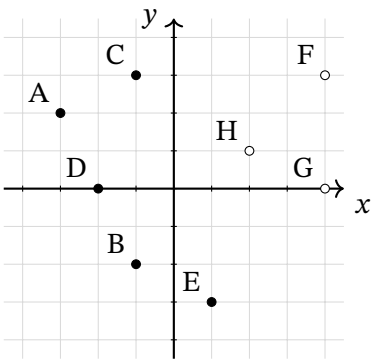
Problem 7: Clustering

The purpose of this problem is to reinforce algorithms and mathematical reasoning surrounding minimum spanning trees.

Clustering is the problem to group similar data points, helping uncover patterns or structures in large datasets, sometimes regarded as a form of unsupervised machine learning. It has broad applications in fields like marketing (for customer segmentation), computational biology (clustering gene expression), image processing (segmenting areas), and many other domains. In this problem, you will write a particular clustering algorithm based off of minimum spanning trees.

The input is a list $A[1 \ldots n]$ of points in some space, a distance function $d(p_1, p_2)$ (that satisfies $d(p_1, p_2) = d(p_2, p_1)$ and $d(p, p) = 0$), and an integer k, the number of clusters. (If it is easier for you, feel free to imagine the 2D plane, where the distance between $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ is given by $d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$, but the algorithm should be basically the same for any distance formula.) The *spacing* of a clustering is defined to be the minimum distance between two points in different clusters. The goal is to partition the points into k clusters that maximizes the spacing.

For example, in the picture below with $k = 2$ and clusters $\{A, B, C, D, E\}$ and $\{F, G, H\}$, the spacing is $\sqrt{13}$, achieved between C and H. This example does maximize the spacing.



1. Before applying a graph algorithm to solve this problem, your must turn the list of points A into a graph. What vertices and edges would you like to create?

2. Describe an algorithm to solve this clustering problem, either in pseudocode or by describing how to modify an MST algorithm discussed in class. What is the running time of your algorithm in terms of n, the number of points? Assume k is a constant. You do *not* need the best running time to get an "E" for this problem, but it should be based on some MST algorithm.

   (Hint: What is a pair of vertices that *must* be in the same cluster, if the goal is to avoid close points being split across different clusters? Does this remind you of an MST algorithm?)

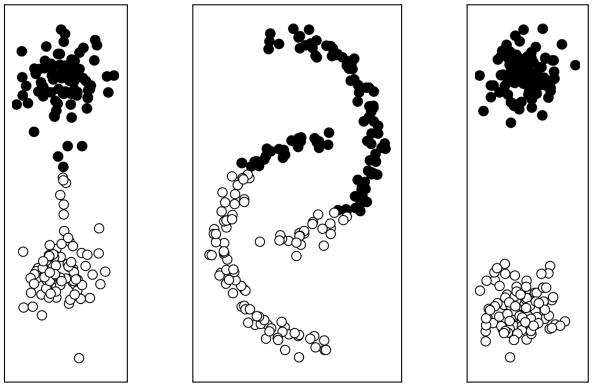
3. This part and the next will guide you through proving the validity of your algorithm (that the output meets the specification). Let $\mathcal{C}$ be your clustering and $\mathcal{C}'$ be any other clustering different from yours. Why must there be two vertices that are in the same cluster in $\mathcal{C}$, but in different clusters in $\mathcal{C}'$?

4. Let d be the spacing (minimum distance between two points in different clusters) of your solution $\mathcal{C}$. Use the previous part to show that $\mathcal{C}'$ must have spacing at most d.

Problem 7X: Clustering (Extensions)

This is an extension problem that builds on the ideas of Problem 7. **Pick one** of the following to complete:

1. Read the following blog post that describes the k-means clustering algorithm, another commonly used algorithm for a slightly different notion of "cluster". This algorithm is commonly taught in machine learning classes. Below are a few examples of the results of a k-means clustering algorithm. Think about whether or not your MST-based clustering algorithm from Problem 7 would produce the same clusters.



Join the discussion on Canvas by responding to the following prompt, raising other questions of your own choosing, or replying directly to other students' responses. (suggested 150–300 words)

> Clustering is a fundamentally subjective problem. Every clustering algorithm defines the notion of "best cluster" slightly differently, and thus the choice of *which* algorithm, the number of clusters, the methods of data preprocessing (if any), etc. are all decisions that you, as the engineer or scientist, must decide and justify.
>
> How might you decide these parameters when you need clustering in applications? It may help to consider: example situations where clustering could be useful, when each algorithm could be misleading, and how easily each form of clustering can be communicated to non-technical audiences.

You are *not* required to cite the article or any other sources, but be sure to discuss at least one difference between the two algorithms. If you discuss any features or theorems regarding k-means, MST-based clustering, or example applications of clustering that are not in the lecture slides or this reading, please cite your sources. **Resubmissions will not be available for this part.**

2. There are many different algorithms for clustering that optimize for different criteria (see the previous extension choice for details), and as humans we also have our own intuition for what a cluster is, that may or may not be captured by any of these algorithms. In this programming assignment, you will implement your clustering algorithm from Problem 7 and evaluate how its output aligns with your intuition on a variety of test inputs.

Implement and run your clustering algorithm on the test sets of points provided. We have included starter code for a graph data structure, and GUI code that that plots your algorithm's output in a Java applet. Look through these tests and determine for which tests the algorithm meets your intuitive understanding of clustering, and for which tests the algorithm produces a clustering that feels unnatural. Screenshot the ones that feel unnatural and place them in a separate document. For each one, briefly explain how you would prefer to cluster it differently. This is your opinion, so there are no right answers. As a bonus (no effect on grading), can you think of any ways to preprocess the data for your algorithm to get more intuitive clusters?

Submit both your code and your mini-report PDF together on Gradescope, which is linked in Canvas. Provided starter code is available both on the course website and Canvas. Do not use libraries beyond the standard Java API. Your code will be autograded, but your mini-report will be manually graded.

Problem 8: Contact tracing...ish?

Last week, you developed an algorithm for contact tracing during a social event in a pandemic. The organizers held such social events multiple times over several days, with the same group of people each time, but each person potentially having different close contacts during each event. You have this data that says who were close contacts on which days. Luckily, no one ever got sick. But today, you're being contracted to use this data to figure out something else.

Apparently, participant s has been spreading rumors that Taylor Swift is going to divorce Travis Kelce! Make the following assumptions about the situation:

- It is impossible to overhear the rumor if you are not a close contact of someone who is talking about the rumor.
- No one hears this rumor from anywhere outside this event.
- Other than these two rules, anyone who knows about the rumor can gossip about it with others. In particular, listeners who hear a rumor can gossip with their own close contacts during the same event.

You are being contracted by Taylor and Travis to identify everyone who could potentially have heard the rumor, so that you can contact them and definitively shut down the rumor. At the same time, you don't want to needlessly contact people if it was impossible for them hear the rumor.

Formally, you have a list $A[1 \ldots m]$ of triples (p, p', t) (with $p \neq p'$), meaning that person p and person p' were close contacts on day t, and a starting participant s who began spreading the rumor on day 1. Assume that this list is given to you sorted by t. Assume $1 \leq p \leq n$ and $1 \leq p' \leq n$ (there are n people total) and $1 \leq t \leq k$ (there are k days). Your goal is to output all people who potentially know the rumor on day k. For a score of "E", your algorithm should run in time at most $O(n + m)$, independent of k. A correct algorithm that achieves a worse running time can receive up to an "S" grade.

1. Describe how you would construct a single graph with vertices and edges to model this situation. How many vertices and edges are in your construction (in big-O)? How long does it take to construct them?
2. Which graph algorithm(s) would you like to run on your graph in order to identify everyone currently infected? What do you do with the result of the graph algorithm(s) to solve the problem? How much time does this take?