Questions and Exercises to work out and turn in:

Grading Guidelines:

* A right answer will get full credit when:

1. It is right (worth 25%)
2. It is right **AND** neatly presented making it easy and pleasant to read. (worth an **extra** 15%)
3. There is an **obvious and clear link** between 1) the information provided in the exercise and in class and 2) the final answer. A clear link is built by properly writing, justifying, and documenting an answer (worth an **extra** 60%).
4. Calculation mistakes will be minimally penalized (2 to 5% of full credit) while errors on units will be more heavily penalized.

You are welcome/encouraged to discuss exercises with other students or the instructor. But, ultimately, **personal** writing is expected.

* USE THIS FILE AS THE STARTING DOCUMENT YOU WILL TURN IN. **DO NOT DELETE ANYTHING FROM THIS FILE:** JUST **INSERT** EACH ANSWER **RIGHT AFTER ITS QUESTION/PROMPT**.
* IF USING HAND WRITING (STRONGLY DISCOURAGED), **USE THIS FILE** BY CREATING SUFFICIENT SPACE AND WRITE IN YOUR ANSWERS.
* FAILING TO FOLLOW TURN IN DIRECTIONS /GUIDELINES WILL COST **A 30% PENALTY.**

**Objectives of this assignment**:

* to use and manipulate the concepts presented in this module
* to propose and write algorithms in pseudocode
* to analyze the time complexity of algorithms
* to analyze the space complexity of algorithms
* to learn autonomously new concepts

What you need to do:

Answer the questions and/or solve the exercises described below.

Exercise 1 (50 points) Kruskal’s Algorithm

Consider this graph G=(V, E, w) provided as an adjacency-matrix. V = (r, s, t, u, v, w, x, y)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | r | s | t | u | v | w | x | y |
| r |  | 21 | 25 |  |  |  |  | 17 |
| s | 21 |  |  | 24 |  |  |  |  |
| t | 25 |  |  | 27 |  |  |  |  |
| u |  | 24 | 27 |  | 29 | 34 |  |  |
| v |  |  |  | 29 |  | 25 | 21 |  |
| w |  |  |  | 34 | 25 |  | 25 |  |
| x |  |  |  |  | 21 | 25 |  | 15 |
| y | 17 |  |  |  |  |  | 15 |  |

1. (5 points) Draw this graph. If needed, you can draw by hand, take a picture and insert it. Just make sure the drawing is neat and pleasant (neatness is worth 15%)

A diagram of a diagram

Description automatically generated

1. (45 points) Trace **Kruskal’s** algorithm and **show step by the step** the construction of the minimum spanning tree. **Draw** the MST each time you add an edge. **Highlight** the latest added edge with its weight**. Explain/justify** only how you select the first three edges.

The first edge is decided by looking at the adjacency list and seeing which weight is the least for the whole graph. That would be (x,y) with a weight of 15. That is then added to the minimum spanning tree portion A.

A diagram of a diagram

Description automatically generated

Then again we check the adjacency list to see the next lowest weight edge which would be (y,r). So now we add the point r to our minimum spanning tree portion A = {x, y, r} after comparing.

A diagram of a diagram

Description automatically generated

After this we again look at the graph and see which weight is the least of the remaining Q and both (r, s) and (v, x) have the same weight of 21. We will choose (r, s) as the next edge to add to the portion of the complete minimum spanning tree A = {x, y, r, s} after this step.

A diagram of a diagram

Description automatically generated

The remaining steps did not need to be explained per the instructions.

A diagram of a graph

Description automatically generated with medium confidence

A diagram of a diagram

Description automatically generated

A diagram of a diagram

Description automatically generated with medium confidence

Exercise 2 (50 points) Prim’s Algorithm

Consider this graph G=(V, E, w) provided as an adjacency-matrix. V = (r, s, t, u, v, w, x, y)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | r | s | t | u | v | w | x | y |
| r |  | 21 | 25 |  |  |  |  | 17 |
| s | 21 |  |  | 24 |  |  |  |  |
| t | 25 |  |  | 27 |  |  |  |  |
| u |  | 24 | 27 |  | 29 | 34 |  |  |
| v |  |  |  | 29 |  | 25 | 21 |  |
| w |  |  |  | 34 | 25 |  | 25 |  |
| x |  |  |  |  | 21 | 25 |  | 15 |
| y | 17 |  |  |  |  |  | 15 |  |

1. Draw this graph (It is the same as the previous question. Copy/Paste would be just fine).

A diagram of a diagram

Description automatically generated

1. (45 points) Trace **Prim’s** algorithm starting from Vertex and **show step by the step** the construction of the minimum spanning tree. **Draw** the MST each time you add an edge. **Highlight** the latest added edge with each weight**. Explain/justify** only how you select the first three edges.

Starting with initialization and with vertex “w” like described. The minimum spanning tree to start is MST = {w}. The first edge will be the lowest weight from the edges connected to w.

A diagram of a network

Description automatically generated

The next edges to consider are (v, u) with a weight of 29 or (v, x) with a weight of 21. We will add the lowest weight vertex to the minimum spanning tree which is now MST = {w, v, x} after this step.

A diagram of a diagram

Description automatically generated

For the third edge we will choose (x, y) as the weight is only 15. Again it will also be added to the minimum spanning tree. MST = {w, v, x , y)

A diagram of a diagram

Description automatically generated

No further explanation was needed for the remaining steps.

A diagram of a diagram

Description automatically generated

A diagram of a diagram

Description automatically generated with medium confidence

A diagram of a diagram

Description automatically generated with medium confidence

A diagram of a diagram

Description automatically generated with medium confidence

1. (5 points) **Compare** the minimum spanning trees obtained by Kruskal’s and Prim’s algorithms, respectively.

Initialization is the first thing different as Kruskal’s always starts with the lowest weighted vertex of its graph. While Prim’s will always start from some arbitrary spot and the go from ascending order of weighted edges and not create any cycle. Although both produce minimum spanning trees. Ideally the practical applications would be the difference of local decisions in Prim’s algorithm. And then global decision making in Kruskal’s. Prim’s algorithm focuses on growing the MST from a starting vertex, while Kruskal’s algorithm sorts all edges and adds them based on weight without starting from a specific vertex.

**What you need to turn in:**

* Electronic copy of this file (including your answers) (standalone). Submit the file as a Microsoft Word or PDF file.
* Recall that answers must be well written, documented, justified, and presented to get full credit.
* How this assignment will be graded:
* A right answer will get full credit when:
* It is right (worth 25%)
* It is right AND neatly presented making it easy and pleasant to read. (worth 15%)
* There is an obvious and clear link between 1) the information provided in the exercise and in class and 2) the final answer. A clear link is built by properly writing, justifying, and documenting an answer (worth 60%).
* Calculation mistakes will be minimally penalized (2 to 5% of full credit) while errors on units will be more heavily penalized.
* You are welcome/encouraged to discuss exercises with other students or the instructor. But, ultimately, personal writing is expected.

**Appendix**: Grading: What is an OBVIOUS and CLEAR LINK?

Here is an example to explain what an **obvious and clear link** is and how we grade your work.

Consider the following problem:

"(100 points) John travels from Auburn to Atlanta in his car at a speed of 60 mph. Leaving at 8am, at what time will John reach Atlanta".

Here are the answers of three students and their scores:

* **Student 1** answers: "9:48am". Student 1 will get 25 points.
* **Student 2**answers : "John will reach Atlanta at 9:48am". Student 2 will get 25+15 = 40 points
* **Student 3** answers: "The time t to travel a distance d at speed v is equal to d/v = d/60mph. The problem does not provide the distance d from Auburn to Atlanta. Based on GoogleMaps, the distance from Auburn to Atlanta is approximately 108 miles (**document is attached**).



Therefore, the time t = 108 miles/60mph \* 60 minutes/hour= 108 minutes. Since John left at 8am, he will then reach Atlanta at 8am + 108 minutes = 8 am + 60 minutes + 48 minutes = 9:48".

**Student 3** will get 25 + 15 + 60 = 100 points

Do you see the **direct** **link** going from the data provided in the question to the final answer, using general knowledge/formula and documents?.... Can you now solve the following problem and get 100 points?

"(100 points) Alice travels from Auburn to Atlanta in her car at a speed of 60 mph. Leaving at 8am, at what time will Alice reach Atlanta assuming that she had a flat tire that delayed her 30 minutes".