Math F307 Spring 2017

Homework Exercises

Instructions for writing up homework.

- Although you are encouraged to work with your classmates on the homework assignments, you must write up the solutions individually.
- Read the section!
- Use pen or non-smeary pencil. Do not have little scritchies on the side.
- Staple your homework.
- Write legibly, and leave lots of whitespace. Make sure your writing is dark enough to be readable. If this is problematic for you, consider typing your homework.
- Answer the question in complete sentences, where appropriate.
- **Write the problem statements as well as the answers.** This is not optional. Points will be deducted if you do not do this.
- Have headings indicating the section.
- **In the upper right hand corner of your top page,** write your name (first and last) on your assignment, along with the course number (Math 307) and which assignment it is.
- **You are expected to ask questions in class about the homework problems!**

Homework Set 8

DUE Friday, March 24 at the beginning of class.

You should **read** each section!

Section 6.6. To turn in: 10, 11, 13

Additional Problems. Also turn in:

(1) Use Kruskal's Algorithm to find a minimal distance spanning tree for the graph shown in Figure 1. Draw in the edges of the spanning tree on a new copy of the vertices. Break ties by choosing edges with a smaller-labelled endpoint. Also provide a list of the edges in the order you added them.

(2) Use Prim's Algorithm to find a minimal distance spanning tree, **starting at vertex 6** for the graph shown in Figure 1. Draw in the edges of the spanning tree on a new copy of the vertices. Break ties by choosing edges with a smaller-labelled endpoint. Also provide a list of the edges in the order you added them.

(3) Use Dijkstra's Algorithm to find a minimal distance spanning tree for the graph shown in Figure 1, **starting at vertex 0**. List all the vertices along with their distance from vertex 0. Also provide a list of edges in the order you added them.

(4) Use Dijkstra's Algorithm to find a minimal distance directed spanning tree for the digraph shown in Figure 1, **starting at vertex A**. List all the vertices along with their distance from vertex A. Are there any unreachable vertices (from A)? Which ones?

(5) List the vertices of the tree shown in Figure 3 using the preorder traversal.

(6) List the vertices of the tree shown in Figure 3 using breadth first search.

(7) Label the vertices of the tree shown in Figure 3 using a topological ordering.

(8) Find a topological ordering of the vertices of the digraph shown in Figure 2, or explain why no such topological sorting exists.

(9) The "while" statement of the topological ordering digraph algorithm includes the statement that we can find a vertex v with no incoming edges. Explain why, if every vertex of a digraph has at least one incoming edge, then the digraph must have a cycle.

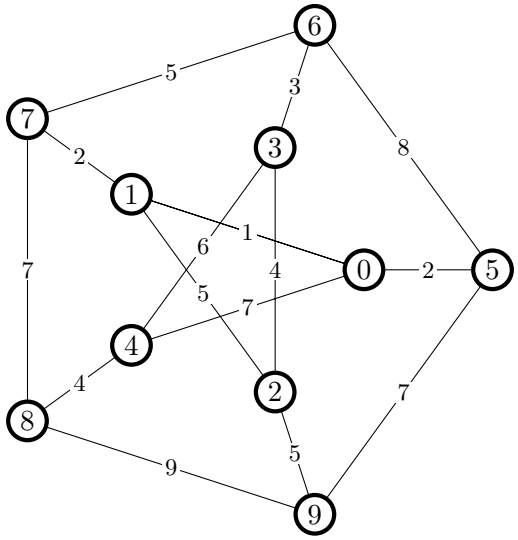

FIGURE 1. Graph for Kruskal, Prim, and Dijkstra's Algorithm

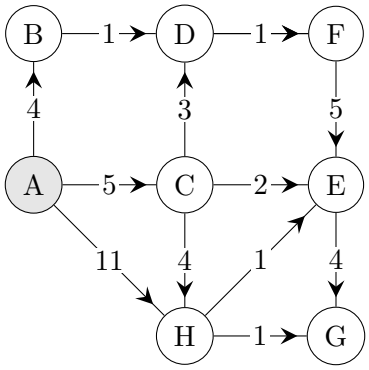

FIGURE 2. A digraph

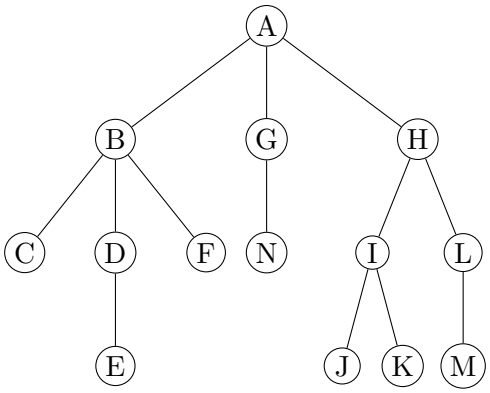

FIGURE 3. A tree