ENGRI 1101	Engineering Applications of OR	Fall 2020	Lab 2

The Shortest Path Problem

Name: _____

Objectives

- Introduce students to the concept of a shortest path tree
- Show students the inner workings of a combinatorial algorithm
- Demonstrate the usefulness of sensitivity analysis in problem solving

Optional Reading Assignment

- Read Handout 3 on the shortest path problem.

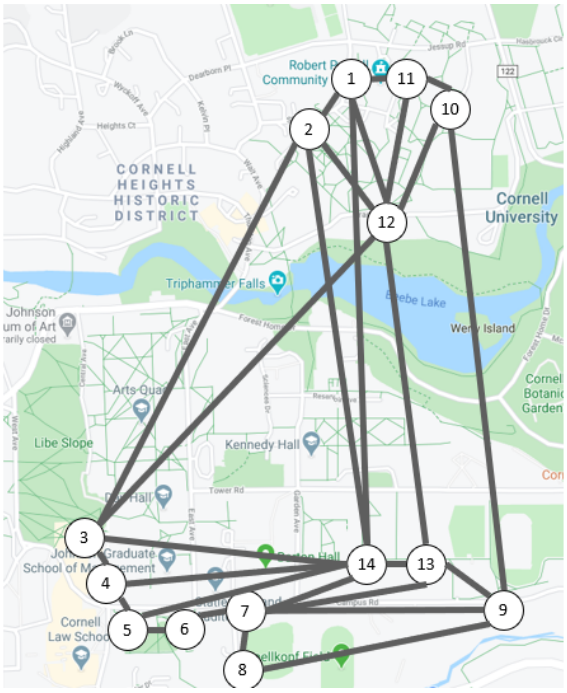
Brief description

In this lab, we review some motivation and observations behind Dijkstra's algorithm for shortest path computation, and analyze how sensitive the solution to the shortest path problem is to changes in the input data.

Part #1: The shortest path problem, and Dijkstra's algorithm

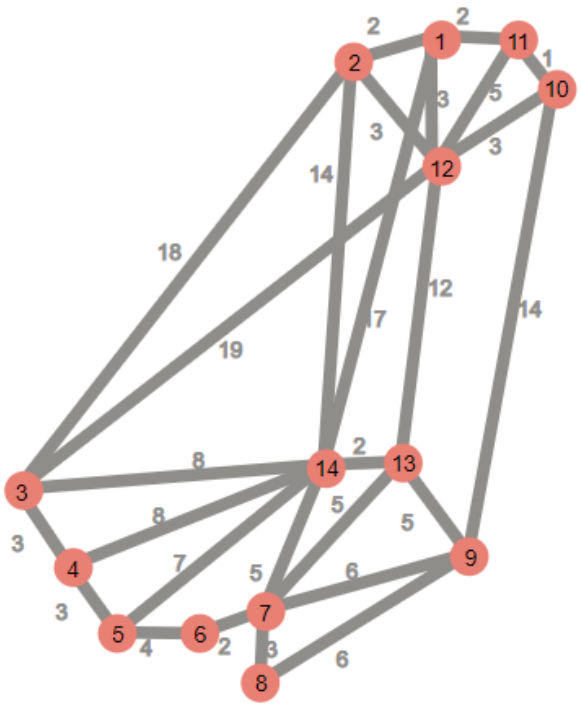
As an Entrepreneurial first-year student, you plan to start a pizza delivery company: Good Pizza, Inc. You will base Good Pizza out of the kitchen on your floor of Jameson Hall and deliver pizzas via bike. You want to guarantee delivery anywhere on campus in 15 minutes or less; in order to maximize the amount of your tips, you want to deliver as many pizzas as you can and deliver them as fast as you can. As a first-year, however, you're still figuring out your way around campus. You've quickly realized that, because of pedestrian traffic, lights, obstacles, and placement of bike stands it is not always best to take the route that covers the shortest distance. Instead, you would like to know the quickest way of getting from the your dorm room to various parts of campus.

To help you in finding the quickest routes, you map the travel time between some of the most-happening locations on campus. Below is an attached map showing the key points on campus. There are several copies of this graph at the end of the lab, which may be useful throughout.



1. Jameson Hall
2. Clara Dickson Hall
3. Willard Straight (and Okenshields)
4. Cornell Health
5. Hollister Hall
6. Duffield Hall
7. Gates Hall
8. Rhodes Hall
9. Cornell High Energy
10. Holland International Living Center
11. High Rise 5
12. Helen Newman Bowling Lanes
13. Lindseth Climbing Center
14. Lynah Hockey Rink

The next page shows a stylized graph of the map above indicating travel times. Note that the positions of some nodes have been slightly modified on the graph so that it is easier to view the information.



Let's load up this graph! Go to `http://engri1101.orie.cornell.edu/` and click `Dijkstra's Shortest Path`. You should see a digital copy of the above graph, and tables set up to run Dijkstra's algorithm.
Your company is based out of Jameson Hall, which is node 1. We'll treat this node as our source s.

What is the node that is closest to Jameson Hall (not considering Jameson Hall itself)? In case of ties, choose any of them. Can there be a quicker route to get to this destination (call it point X) than just to go directly there from Jameson Hall? Why or why not?

Explain why the predecessor ("prev(X)" from class) of X must be the Jameson Hall.

For now on, let X be High Rise 5 (X has ID 11). On the web app, first click the node labelled 1 to start running Dijkstra's algorithm. Then select node 11.

After clicking those two nodes, you should notice that several things happen:

- Node 1 and 11 turn solid blue, and the edge between them darkens; this edge has been "added to the shortest path tree."
- The tables on the right update, and you can see the current version of the Dijkstra table (iteration 2). You can also compare this to the previous table (for iteration 1), faded out below it.
- In the current table, node 11 is now starred and prev(11) is marked as node 1.

The graph shows styles of nodes:

- Fully blue nodes (1 and 11)
- Fully red nodes (e.g. nodes 3 and 9)
- Red nodes with a blue perimeter (e.g. nodes 2 and 14).

What does each style of node represent?

How do we find the next edge to add to the tree (i.e., the next edge to "darken")? One way is as follows. Compute the travel times for all routes that either go directly from Jameson Hall to a destination or go from Jameson Hall to point 11 and then directly to a destination. Take the shortest of all these routes and add it to the tree. For instance, in this case, we consider the following routes. Node IDs are in parenthesis.

Route	Travel Time
Jameson Hall (1) - High Rise 5 (11)	**2**
Jameson Hall (1) - Clara Dickson Hall (2)	2*
Jameson Hall (1) - Helen Newman (12)	3
Jameson Hall (1) - Lynah Hockey Rink (14)	17
Jameson Hall (1) - High Rise 5 (11) - Holland International Living Center (10)	3
Jameson Hall (1) - High Rise 5 (11) - Helen Newman (12)	7

Consider all of the nodes we can reach by these paths (e.g. 1, 11, 2, 12, 14, and 10). What is true about those nodes – and only those nodes – in the web-app table for iteration 2?

There are two paths to node 12 listed above. Look at node 12 in the table for iteration 2. Which of those two paths is indicated in the table? Using just the table – without referring back to the original graph or edge lengths – how can you identify this path?

Now back to Dijkstra and we pick Clara Dickson. Jameson Hall to Clara Dickson is the shortest route on this list (we've marked it with a *). Can there be a shorter route to Clara Dickson than the one that we just added? Why or why not? Answer this question in as much generality as possible. Hint: suppose that there were some shorter path that went from Jameson – SOMEWHERE ELSE – Clara Dickson. How would the length of the first edge in this path have to compare to the length of the edge directly between Jameson and Clara Dickson?

Now we update the table for the next iteration: if two entries have the same destination we drop the more expensive one (e.g., we drop the Jameson Hall - High Rise 5 - Helen Newman route, since we it takes longer than the direct Jameson Hall - Helen Newman route). Then for each destination that can be reached from Clara Dickson, we compute the length of the path using only nodes already marked, with Clara Dickson being the second-to-last node on the path. Fill in the missing entries in the table below. Then, as before, cross out any redundant paths (like the Jameson Hall - High Rise 5 - Helen Newman path which we previously dropped).

Route	Travel Time
Jameson Hall (1) - High Rise 5 (11)	**2**
Jameson Hall (1) - Clara Dickson Hall (2)	**2**
Jameson Hall (1) - Helen Newman (12)	3
Jameson Hall (1) - Lynah Hockey Rink (14)	17
Jameson Hall (1) - High Rise 5 (11) - Holland International Living Center (10)	3
Jameson Hall (1) - Clara Dickson Hall (2) - ...	...
Jameson Hall (1) - Clara Dickson Hall (2) - ...	...
Jameson Hall (1) - Clara Dickson Hall (2) - ...	...

On the web app, click node 2 (when we click node 2, we "mark it." This marking is encoded in the graph when the corresponding node turns blue, and with a * in the updated table for iteration 3). Use the updated table for iteration 3 to verify your work.

Choose the next edge to be added to the shortest path tree. List both the next node to click and the corresponding edge that will be darkened.

Dijkstra's Algorithm continues in this manner. In the next step, we compute the travel times to destinations not already marked (i.e. starred/fully blue) that are adjacent to the lastly marked node, using routes that involve only already marked nodes as intermediate steps. Do the next iteration of the algorithm by hand based on your previous answer: write down the nodes that get marked and/or have their labels updated in this iteration. It will help to look at the map attached to the end of this lab (you can rip off the last page and use it as a reference throughout the lab).

Now trace the execution of Dijkstra's algorithm with the software up to the point that you have already computed by hand: at each step, click the next node to mark on the software. You should be able to readily identify this node based off the table shown on the web app.

Now highlight the shortest path tree you just found on the copy of the map at the end of this packet. How can you read out the shortest path from Jameson Hall to other places in on campus from the shortest path tree?

What is the shortest path (both the route and length) from Jameson Hall to Rhodes Hall (node 8)?

Is the 15 minute guarantee a good one? What about a 30 minute guarantee, assuming some reasonably small fluctuations in the actual travel time due to traffic, etc.

Is the shortest path always unique? If yes, why? If no, can you give an example of two nodes in the graph for this lab that have more than one shortest path between them?

Part #2: Sensitivity analysis

The next several questions ask you to analyze how sensitive your output is to the precise data that has been used. This is called "performing sensitivity analysis" for the input.

Suppose that there was a parade on campus so that the time to go from Holland International (10) to Cornell High Energy (9) *directly* was increased to 20 minutes.

Would this change the shortest path tree? How would the shortest path tree change?

Can such an increase affect any of the routes that did not originally include the leg from Holland International (10) to Cornell High Energy (9)? Why or why not?

Now verify your answers! Click the "double arrow" button below the graph (the button on the left, with two arrows in different directions). This feature will allow you to change the length of an edge: on the first prompt enter 10, on the second prompt enter 9, and on the third prompt enter 20. The edge between nodes 9 and 10 should now have cost 20. Rerun Dijkstra's algorithm (feel free to click the fast forward button). Did you make any mistakes?

Suppose that travel time from node (10) to node (9) increased from 14 minutes, but not all the way to 20 minutes. By how much could the travel time increase **without** the shortest path route to node (9) changing (i.e. matching the original route found by the web app)?

Hit the reset (circular arrow) button below the graph. Doing so resets any changes to the original edge lengths. Suppose that your competitor, Better-Than-Adequate Pizza, decides to promote their brand by installing a moving walkway that takes students from Clara Dickson on North Campus to the Lynah hockey rink. This has the effect of reducing the travel time from Clara Dickson (2) to Lynah (14) so that it only takes 5 minutes. Would this change alter the shortest path tree? Does this affect any of the routes that did not originally include the leg from Clara Dickson to Lynah? Why or why not? Feel free to check your work by asking the program to change the length of the edge between nodes 2 and 14.

Suppose that all the travel times obeyed the following inequality:

$$\text{(travel time from A to C)} \leq \text{(travel time from A to B)} + \text{(travel time from B to C)}$$

for all nodes A, B and C. This is called the **triangle inequality**. Suppose our input was the complete graph (that is, there is an edge between every pair of nodes). If the travel times on this input obeyed the triangle inequality, what would the shortest path tree look like?

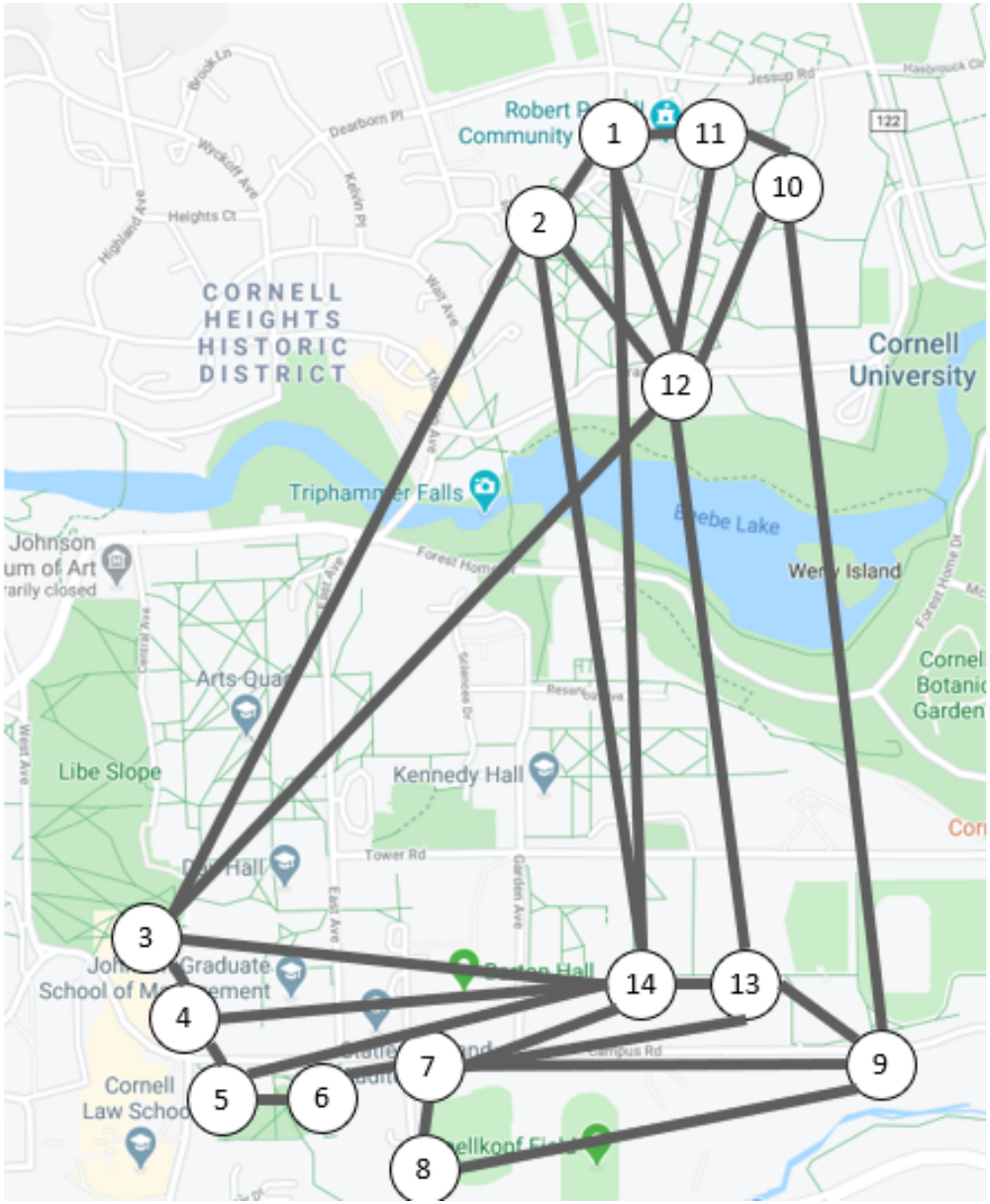
Part #3: NYC Data

Download the entire folder Lab 2 Extension from Canvas. Open the .ipynb file in Jupyter notebook, and work through the Juypter notebook. Then, answer the following questions after going through the NYC pizza example.

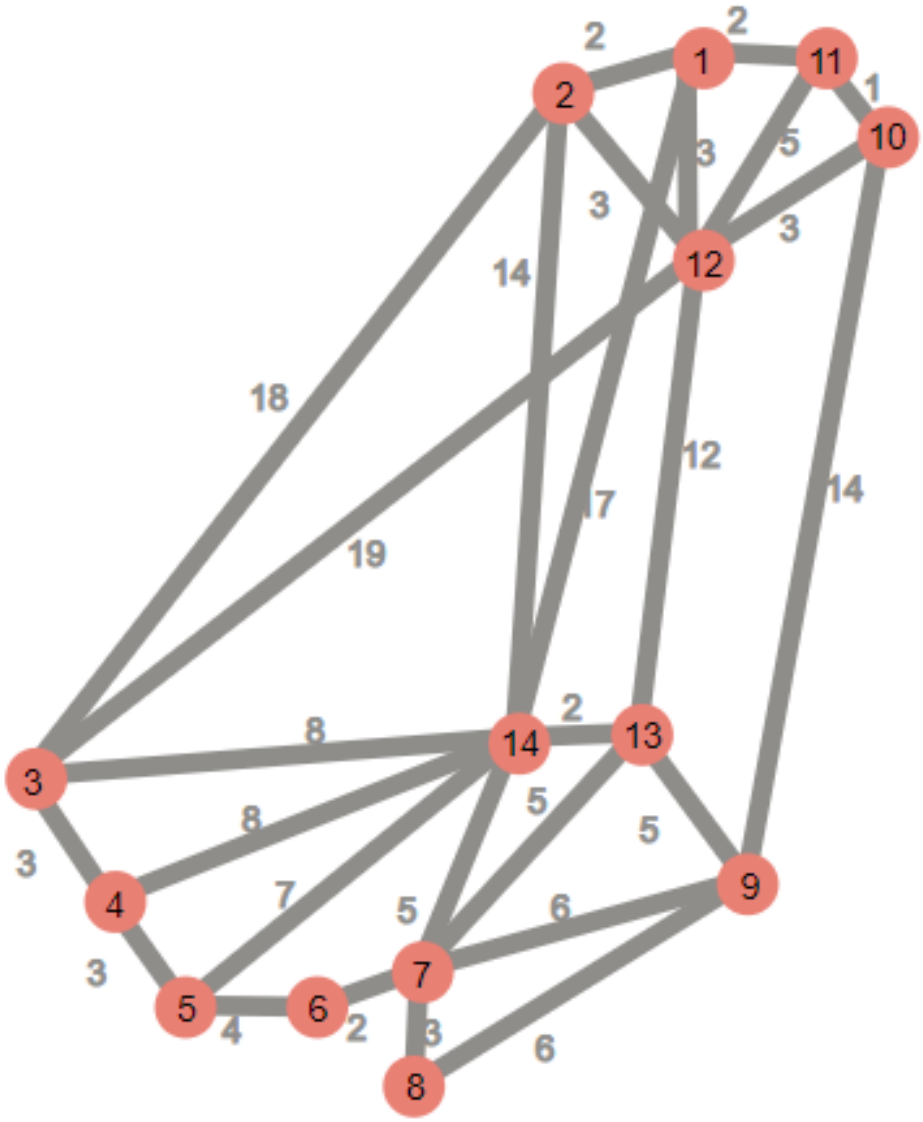
What is the variable *dfl*, and what are its columns?

Without a traffic jam on Queensboro Bridge, is a 30-minute guarantee reasonable?

What would you recommend as a delivery time guarantee to all the marked locations given that there are often traffic jams on Queensboro Bridge?



1. Jameson Hall
2. Clara Dickson Hall
3. Willard Straight (and Okenshields)
4. Cornell Health
5. Hollister Hall
6. Duffield Hall
7. Gates Hall
8. Rhodes Hall
9. Cornell High Energy
10. Holland International Living Center
11. High Rise 5
12. Helen Newman Bowling Lanes
13. Lindseth Climbing Center
14. Lynah Hockey Rink


1
2
3
4
5
6
7
8
9
10
11
12
13
14
2
2
1
3
5
3
3
14
18
12
14
7
19
8
2
8
5
5
3
7
5
6
3
4
2
3
6