

# Directed Weighted Graph

Dijkstra's Shortest Path Algorithm

# Requirements

Create file in python with a **comment** containing the academic honesty pledge as shown below. Add another, separate comment to the file containing your name

- Write a python program that creates a graph using a textarea and the formatting described in a later slide
- Your code will print out the shortest path in the graph described in later slides.
- Your code should generate both the form (with a textbox) and the output.

```
# I honor Parkland's core values by affirming that I have  
# followed all academic integrity guidelines for this work.  
  
# your name
```

# Input format: This is a **directed & weighted graph**

```
vertex1  
vertex2  
vertex3  
...  
vertexn  
#end
```

the names of the vertices, one per line. NO EMBEDDED SPACES!!!

Ignore duplicates

There are no negative edge weights.

Keyword that shows the end of the vertices

```
vertex1, vertex2, 5.6  
vertex1, vertex3, 8.9  
...
```

the edges in the graph. Format is:

`vertex [comma] [space] vertex [comma] [space] weight [newline]`

When you're out of data, there are no more edges. Ignore invalid edges. Weights can be decimal numbers

# Shortest path

If there are edges labels "START" and "END", perform Dijkstra's algorithm on the graph and print:

- The shortest path from START to END as a series of edges.
- The weights of the edges used. (You'll need to mess with construct path)
- The total weight of the path. (You'll need to mess with construct path OR the distance map)

# Input sample

```
spot1  
place2  
place106  
hallway5
```

```
START
```

```
vertex8
```

```
END
```

```
#end
```

```
START, place2, 5.6
```

```
place2, END, 8.9
```

# Turn in

The code you wrote or modified.

A link to the webpage.