

# Dijkstra's Shortest Path – Live Mobile Graph Builder

By: Caleb Bostic-Gardner

# What is Dijkstra's Shortest Path Algorithm? (SPA)

- Like the name might imply, it's an algorithm for finding the shortest path between nodes in a Graph
- It's a greedy algorithm which means it builds the shortest path by making locally optimized decisions.

# How does the SPA work?

- The algorithm keeps track of three key pieces of information

1. Visited nodes
2. Unvisited nodes
3. The cost to reach a node

## ○ Steps

1. Choose the start node and set its cost to 0
2. Set the distance for all other nodes to Infinity (or very high number b/c they are unknowns for now)
3. Now visit the node with the smallest known distance from the start node
4. Examine the current node's unvisited neighbors
5. Calculate the distance to each neighbor from the start node
6. Update the known distance for a node if the calculated distance is smaller
7. Add the current node to the list of visited nodes
8. Repeat steps 3-7 until all nodes have been visited

# Why SPA's process works

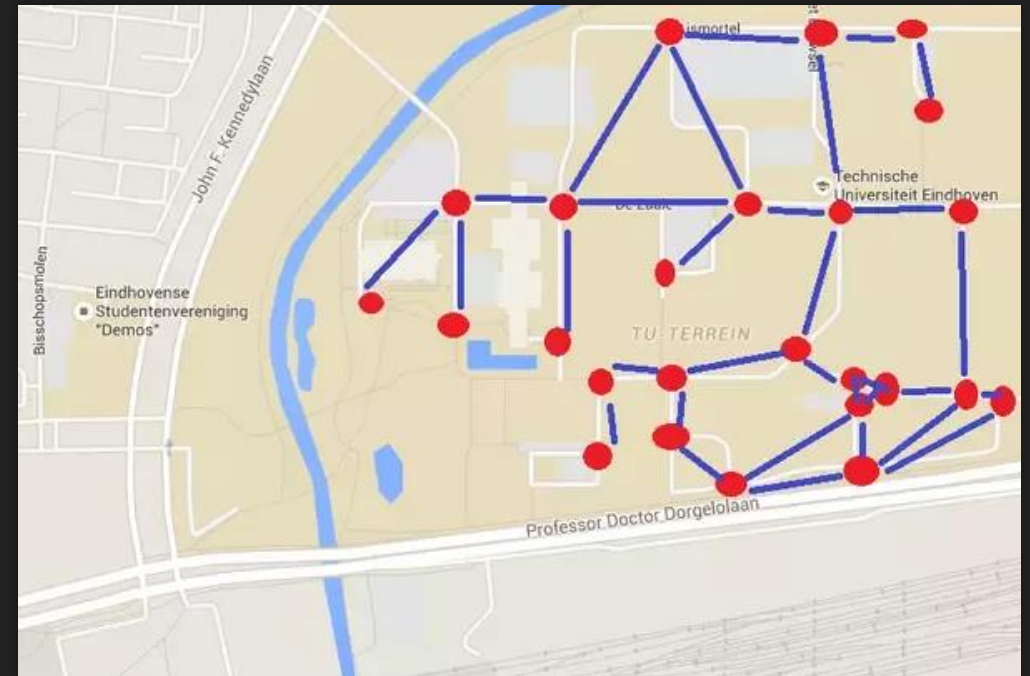
- The algorithm can find the shortest path if there is a valid path between the nodes because it is able to make minimal cost decisions from node to node; incrementally constructing the shortest path
- Because of this behavior, if along the way it picks a path that had the potential to be shorter but ultimately becomes longer, it can backtrack and continue from the last known shortest distance node

# Where does my project fit in?

Wouldn't it be cool to be able to build a graph live?

# Make an app!

- I'm a mobile software developer, so the idea came to me to make an app to visualize it.
- The idea was to automatically drop markers at a set interval, and then save that stretch of plotted markers after some time.
- The saved markers over that span would form a route segment.
- Each segment would have a line drawn with a color to identify it.



Source: <https://codingwithalex.com/introduction-to-graphs/>

# Goals for the project

01

Create route segments live (walking or driving)

02

Extract markers from route segments

03

Build graph from markers

04

Visualize shortest path algorithm

- Draw the path from start marker to end marker

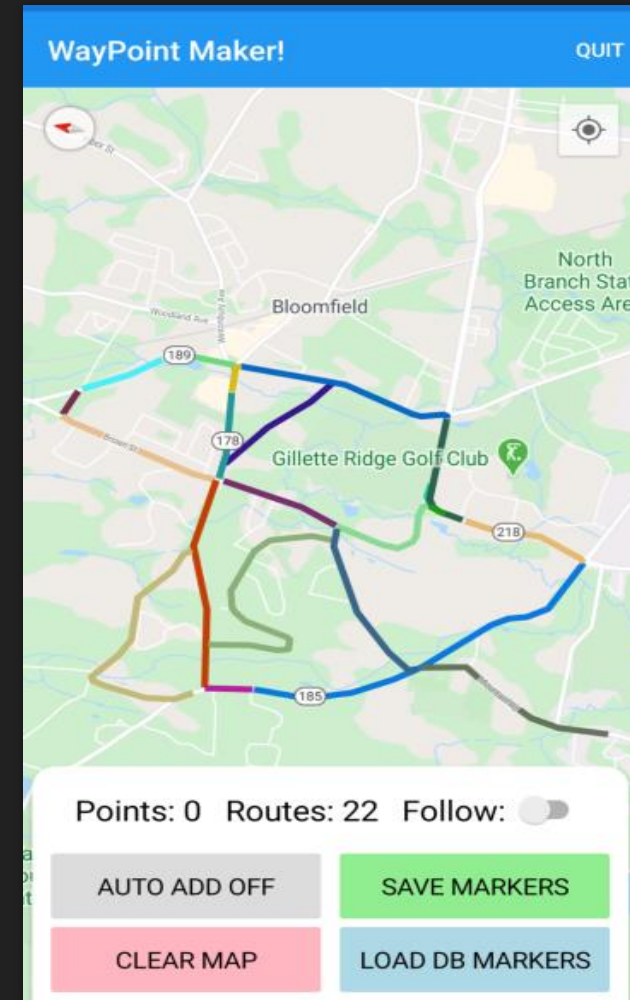
# Accomplished Everything?

Yes... (no)



# How far did I get?

- Everything was do-able, except the “live” part
- App capabilities:
  - Works on Android + iPhone
  - Create, visualize and store route segments + markers
  - Export the saved route segments as JSON via email
- The image on the right is a real screenshot of one of the test runs I did, and the resulting graph of route segments.
- From this I got JSON, Let's see it.



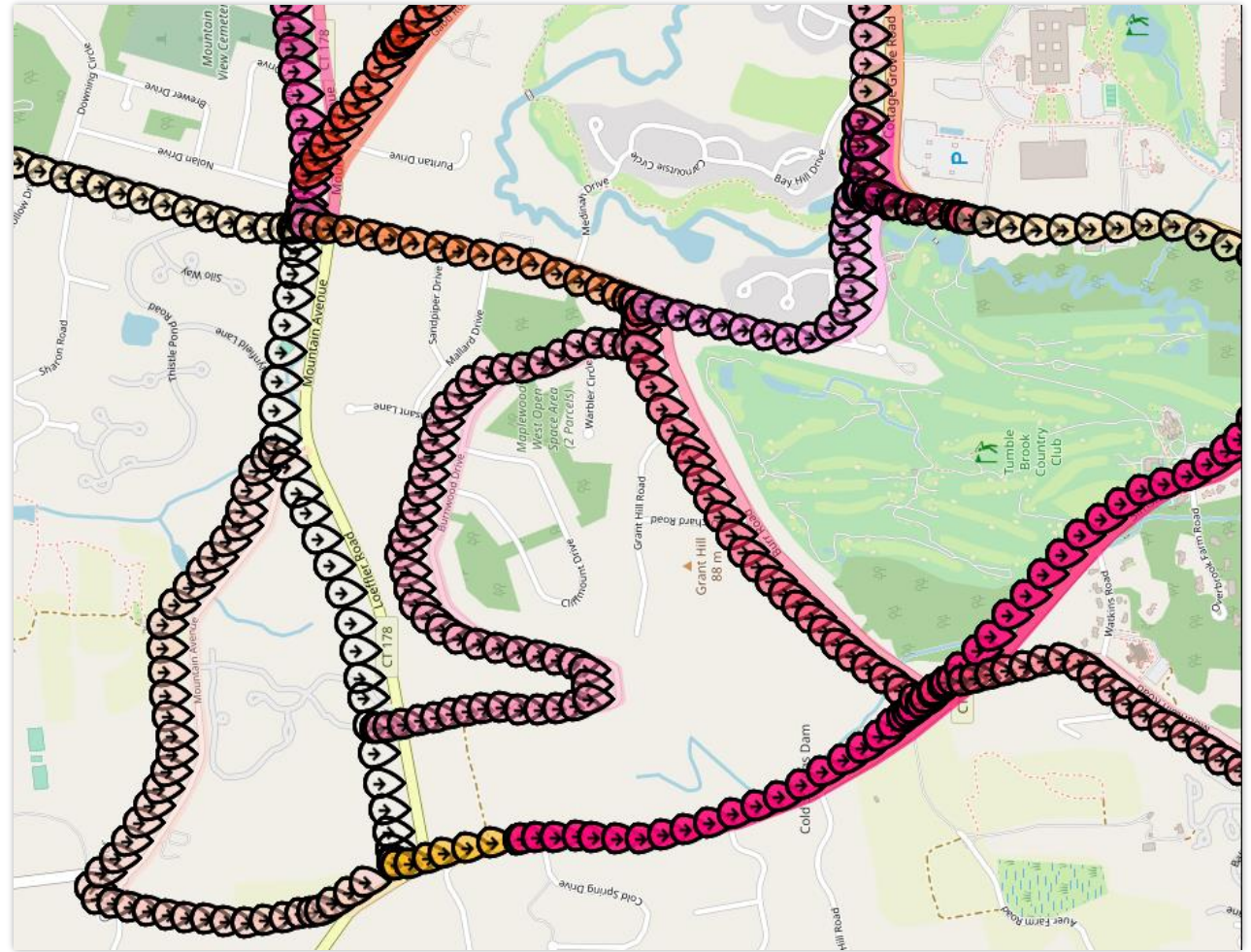
# Scary JSON?

- Hopefully not!
- The 'routes' key in the json object is an array of routes (ironically)
- Each object in the routes array has an id (from database), its color, and its array of route points (markers)
- And each route point just has a longitude, latitude and id (uniquely generated)

```
{
  "export_date": "2020-04-28T01:06:09.75109-04:00",
  "map_center_lat": 41.806065,
  "map_center_long": -72.750758,
  "routes": [
    {
      "Id": 1,
      "Color": "#FF0679E2",
      "RoutePointsBlob": "W3siSWQiOiIrajJhQ2dpQVZVbTVHY1pQIiwiTGF0aX"
    },
    {
      "Id": 2,
      "Color": "#FFE8BA70",
      "RoutePointsBlob": "W3siSWQiOiJvUlV3bnd0S1gwaUtSRXB6IiwiTGF0aX"
    },
    {
      "Id": 3,
      "Color": "#FF29624D",
      "RoutePointsBlob": "W3siSWQiOiJSY0gyMm1H MVZVdVRvYmViIiwiTGF0aX"
    },
    {
      "Id": 4,
      "Color": "#FF0468C7",
      "RoutePointsBlob": "W3siSWQiOiJHNEQ3Uk1ZN2VFbUVWYlBVIiwiTGF0aX"
    },
    {
      "Id": 5,
      "Color": "#FFD5BC18",
      "RoutePointsBlob": "W3siSWQiOiJySWhDd09pMzNrT2YvSmljIiwiTGF0aX"
    },
    {
      "Id": 6,
      "Color": "#FF21989D",
      "RoutePointsBlob": "W3siSWQiOiJySWhDd09pMzNrT2YvSmljIiwiTGF0aX"
    }
  ]
}
```

# What does the JSON do?

- The JSON give us the next piece of the puzzle
- A webpage that gives us a better visualization of the data and a way to filter it
- From the picture on the right each route segment is drawn with its markers and line connecting them



# How does it all connect?

- One feeds into the other!
- The mobile app provides the routes
- The webpage provides a visualization and way to filter all the collected markers to then build the graph
  - Which will be composed of adjacency lists!
- The Java program can then take the markers and each markers neighbors and run the Shortest Path Algorithm



# What's left?

- Getting the Java program to successfully run!
  - Setting up data-structures correctly
  - Creating adjacency list for graph from exported markers

**Thanks!**