# CityHop

Shortest-path algorithms and travel

Road-trips are excellent opportunities to explore.

It’s possible to take a scenic route without going too far off of the direct path.

# Motivation

- Directions are focused on accuracy and traveling the shortest distance
- Focus is not placed on the bigger picture regarding step-by-step directions
- It is easy to miss interesting landmarks by following the quickest route

- By determining the shortest path as-the-crow-flies and without directions:
  - guidance can be provided regarding places of interest
  - it is possible to modify the trip without going too far out of the way

# Features and Constraints

- At least one large city per region is included
  - San Francisco and Sacramento are included, but Oakland and San Jose have be omitted
- All national parks which can be driven to are included or are in the process of being added

# Methodology

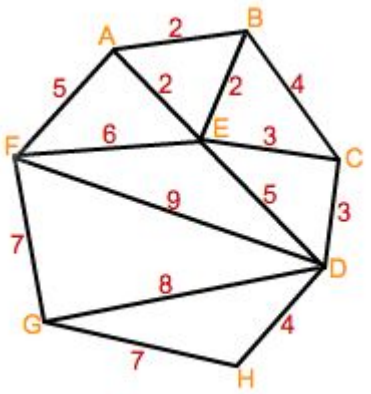
- Google Geocoding API was used to establish start and end coordinates
- The Haversine formula determined the closest locations in the network to the start and end points
  - I opted to calculate distance this way due to the spherical nature of the Earth

- Dijkstra’s algorithm was used to calculate the shortest path from one point to another

# Process

- Geocoding is the process of reconciling addresses/landmarks with latitude and longitude coordinates
- I opted to use the Google Geocoding API, however there are other tools such as the geocoder developed by HERE out of Fargo

- Dijkstra’s algorithm can find the shortest path between nodes in a weighted graph
- The same premise can be applied to cities, or even landmarks in a town



The weight between cities in a network can be determined with latitude and longitude and the Haversine formula for great arc length:

$$a = sin^2 \cdot \frac{lat2 - lat2}{2} + cos(lat1) \cdot cos(lat2) \cdot sin^2 \cdot \frac{lng2 - lng1}{2}$$

$$c = 2 \cdot atan2(\sqrt{a}, \sqrt{1 - a})$$

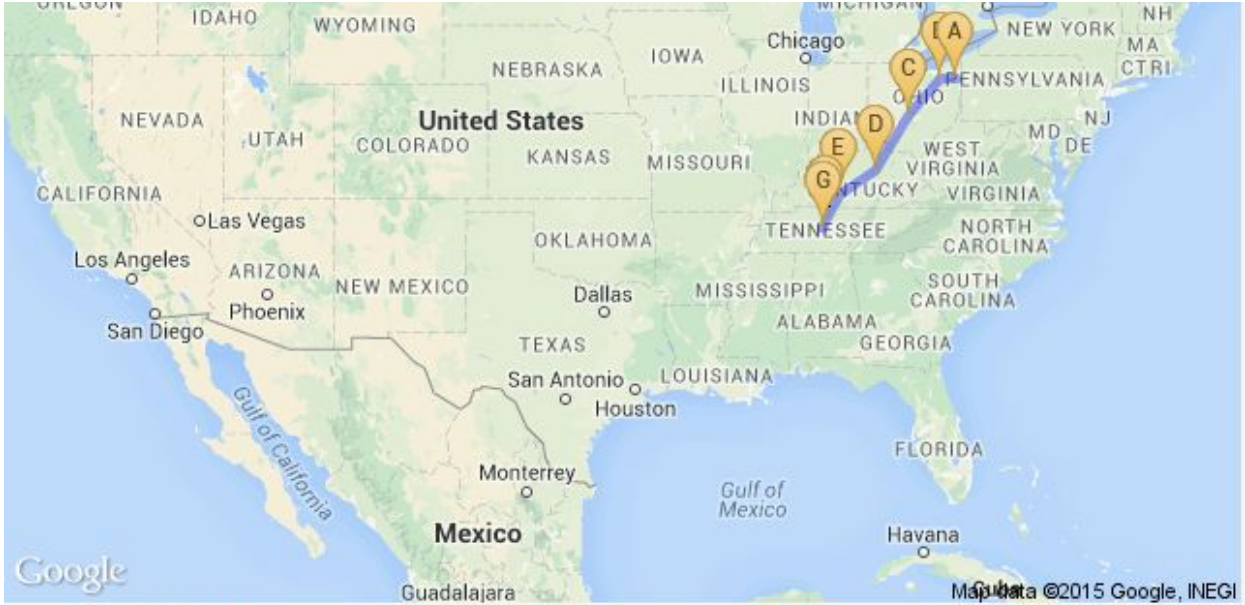
$$distance = radius\ of\ Earth \cdot c$$

- Connections were made based on the following criteria:
  - Large cities closest to one another relative to other cities should be connected
  - Landmarks are connected to either: the closest landmark, the closest city, or both depending on what seems most sensible
  - No points will be connected that span a body of water that is not traversable
    - e.g. you cannot drive across the Gulf of Mexico

- I performed testing to determine whether the landmarks and connections made sense
- Sometimes I had to add or remove landmarks or connections
  - Initially, landmarks were farther apart and the connections had a greater weight
  - As more landmarks were added, the connections with a lower weight were favored as they are more accurate

- I populated a database with values for city/state and corresponding latitude and longitude coordinates
- Geocoding is used to reconcile city/state values with latitude and longitude coordinates, so it is possible to enter full addresses

# Examples




514.0 Miles

A: Warren, OH

B: Cuyahoga Valley National Park

C: Columbus, OH

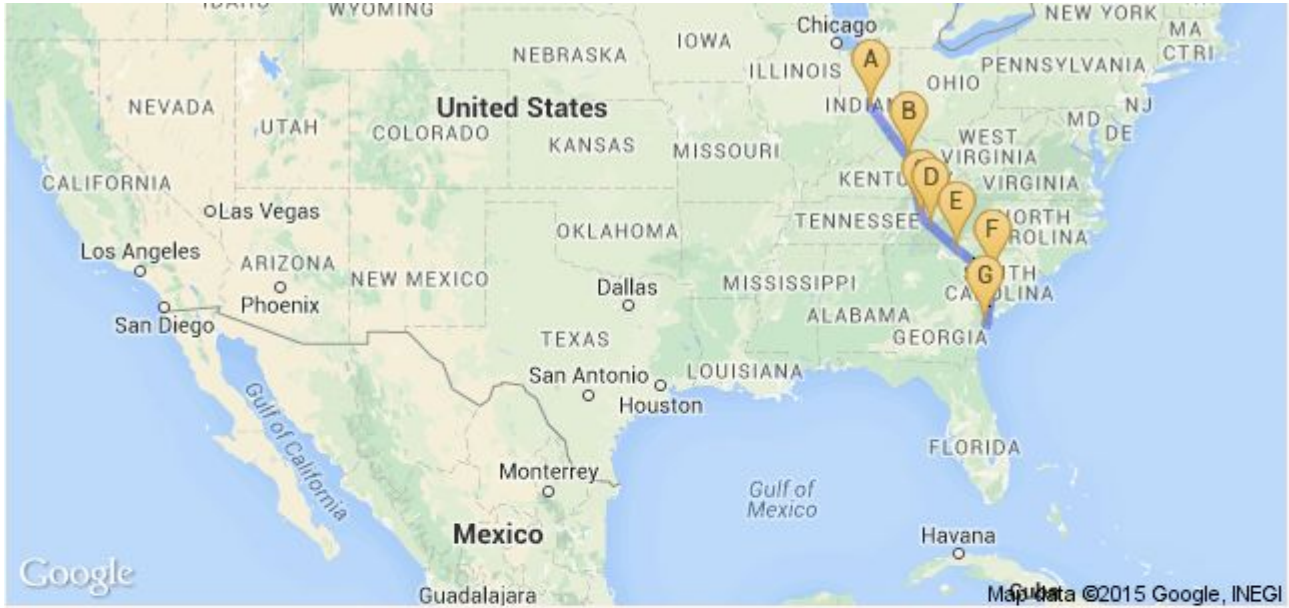
D: Lexington, KY

E: Mammoth Cave National Park

F: Nashville, TN

G: Franklin, TN

NPS site




650.0 Miles

A: Indianapolis, IN

B: Lexington, KY

C: Knoxville, TN

D: Great Smoky Mountains National Park

E: Greenville, SC

F: Congaree National Park

G: Savannah, GA


850.0 Miles

A: Pocatello, ID

B: Provo, UT

C: Grand Canyon National Park

D: Flagstaff, AZ

E: Phoenix, AZ

F: Tucson, AZ

G: Saguaro National Park

H: Sierra Vista, AZ


527.0 Miles

A: Saint George, UT

B: Zion National Park

C: Bryce Canyon National Park

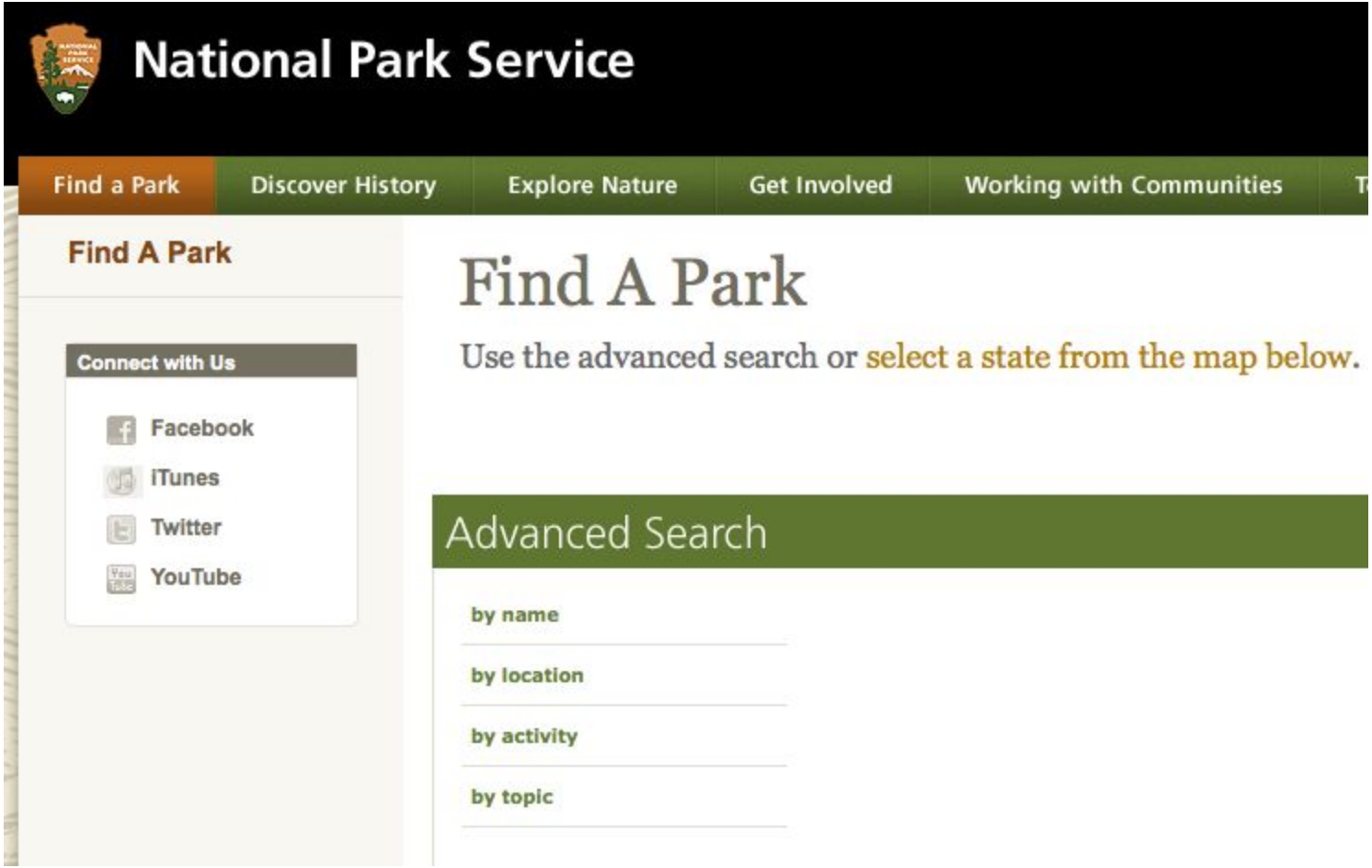
D: Capitol Reef National Park

E: Canyonlands National Park

F: Arches National Park

G: Black Canyon of the Gunnison National Park

H: Denver, CO


National Park Service
Find a Park
Discover History
Explore Nature
Get Involved
Working with Communities
Find A Park
Connect with Us
Facebook
iTunes
Twitter
YouTube
Find A Park
Use the advanced search or select a state from the map below.
Advanced Search
by name
by location
by activity
by topic

# Future Ideas

- Add support for national forests, state parks, and other landmarks
- Suggest second and third shortest paths as alternative routes
- Provide links relevant to landmarks

- Refactor code and improve implementation
  - Possibly use a different GUI package
  - Probably change the presentation altogether
- Use something other than a static map so that zooming is possible
- Make results more interactive
  - Due to difficulties with the GUI package, I was unable to include more than one link; ideally multiple links can be provided

- Incremental distance might be nice to have
- Google map markers only allow a single character; if the route is long, I run out of characters after ‘Z’
  - Include digits
  - Recycle characters
- Determine whether there is a better way to present routes for long trips
  - Collapsible content
  - Legs of journey

- If possible, interactive map markers instead of or in addition to textual description of route
  - Location name
  - Links
  - Other information

- Algorithm to dynamically determine route
  - Pro: the route would be sensible and follow established routes
  - Con: the route would probably avoid scenic, leisurely route
  - Con: this would use considerably more API calls for geocoding and/or many more values would need to be stored in a database

- The purpose of this program is to offer the most optimal sub-optimal route, otherwise most natural landmarks would be skipped due to their distance from main thoroughfares
- The algorithm would probably need to find the optimal route and then try to include landmarks