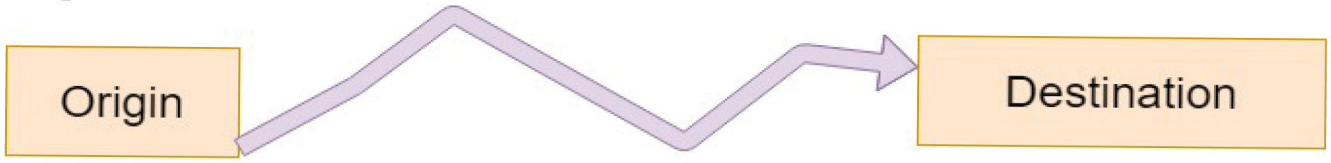
Group: Elena Roosevelt
Elevation based navigation

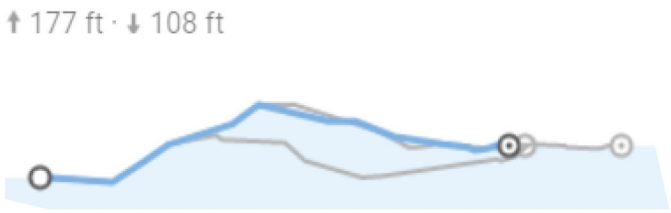
INTRODUCTION: THE PROBLEM

Input:



- Maximizes or minimizes elevation gain
- Limiting the total distance between the two locations to x% of the shortest path

Output:

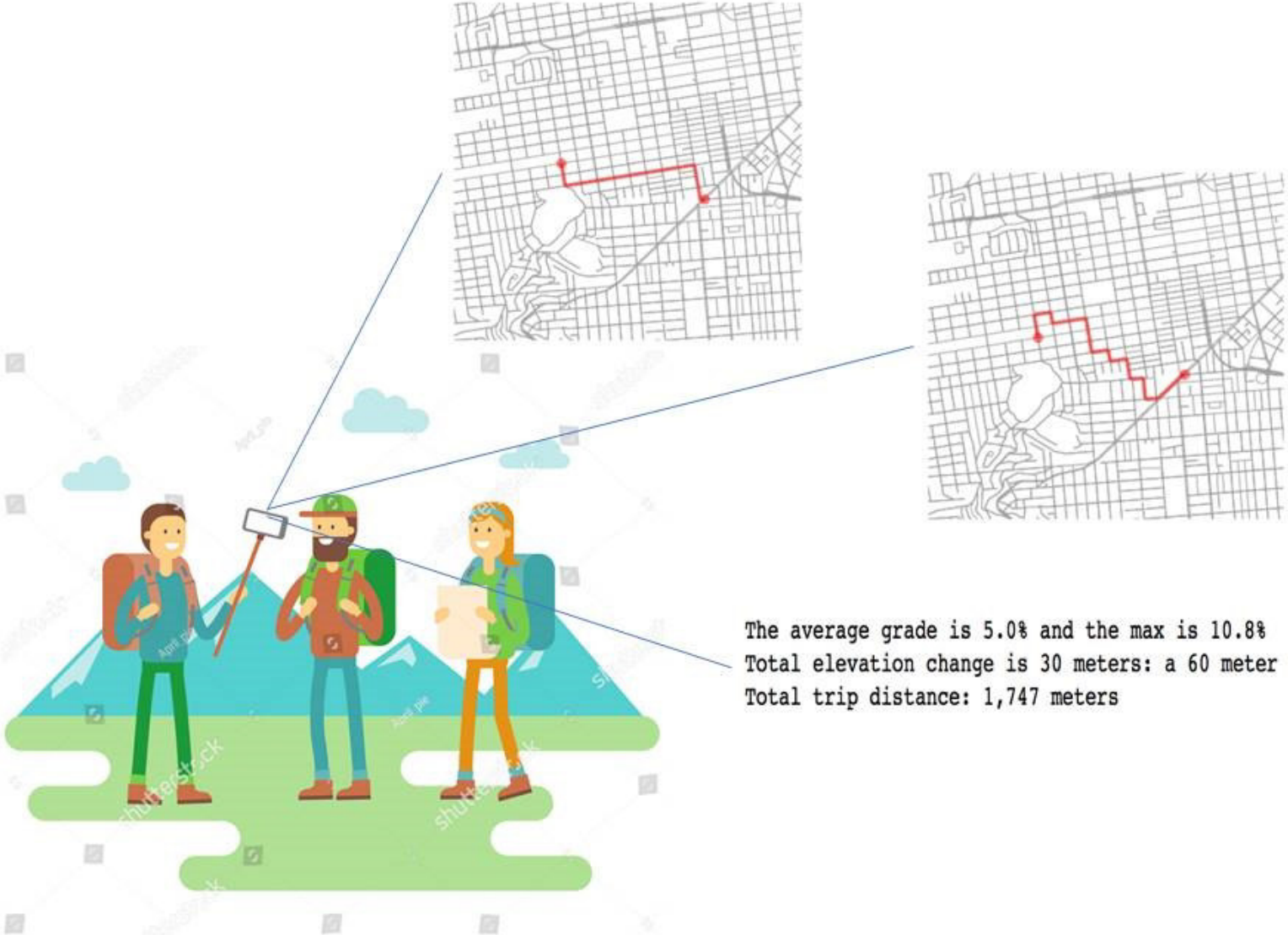


THE SOLUTIONS

Minimize uphills or downhills:
Apply a penalty to the length based on the grade

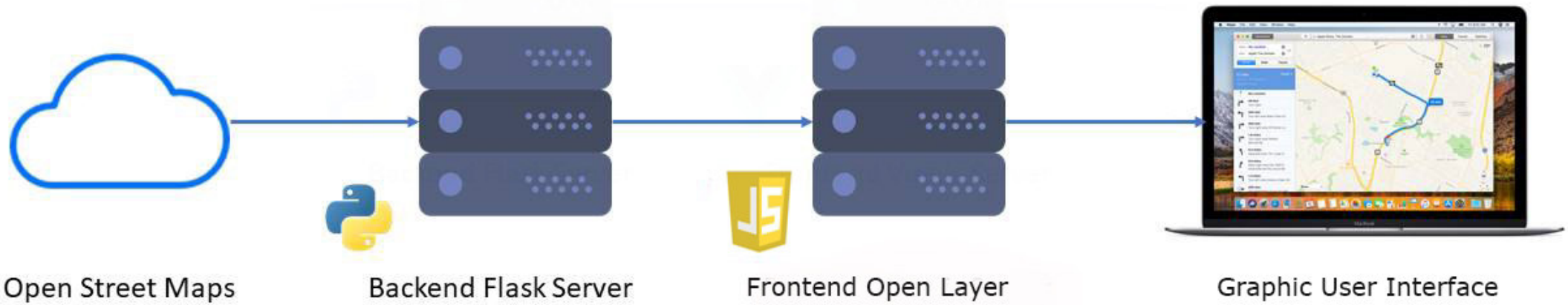
PENALTY = GRADE**2

IMPEDANCE = PENALTY * LENGTH



The average grade is 5.0% and the max is 10.8%
Total elevation change is 30 meters: a 60 meter ascent and a 30 meter descent
Total trip distance: 1,747 meters

ARCHITECTURE



VIEW

Display:

- Maps
- Routes
- Elevation
- Length
- Input boxes
- Buttons

coordinates

MODEL

Converts addresses to coordinates

Updates graphs

Finds optimized route

Returns route

Address: origin
Address destination

Input boxes

Get route button

Find next point button

Reset button

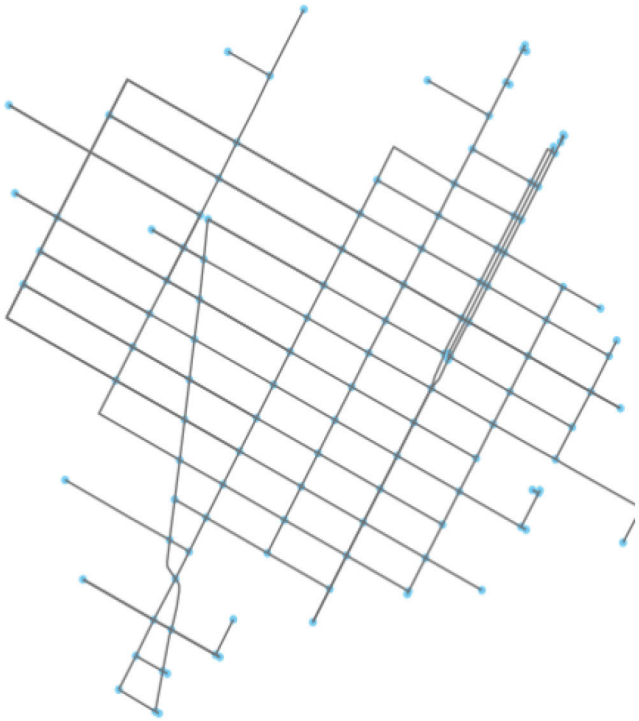
CONTROLLER

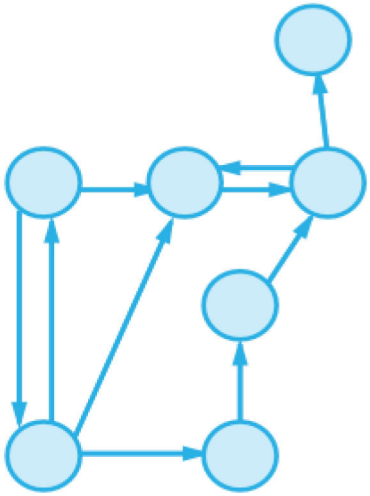
DESIGN

DATA STRUCTURE

Networkx.MultiDiGraph

- Directed graph with self-loops and parallel edges
- Can store multiple edges between two nodes with optional data or value
- Uses dict-of-dict-of-dict-of-dict structure (4 Python dictionaries). Each dict is an adjacency list

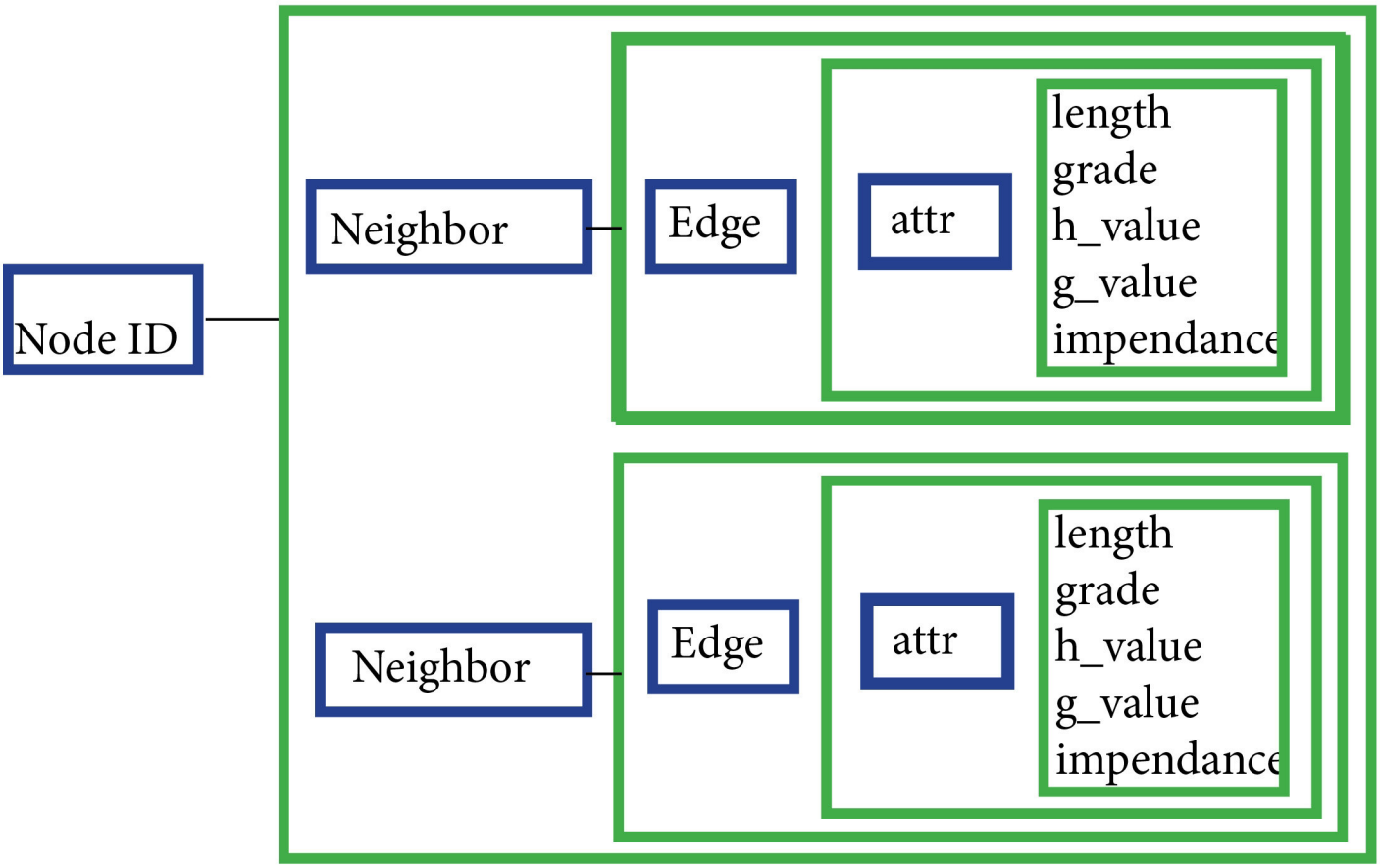


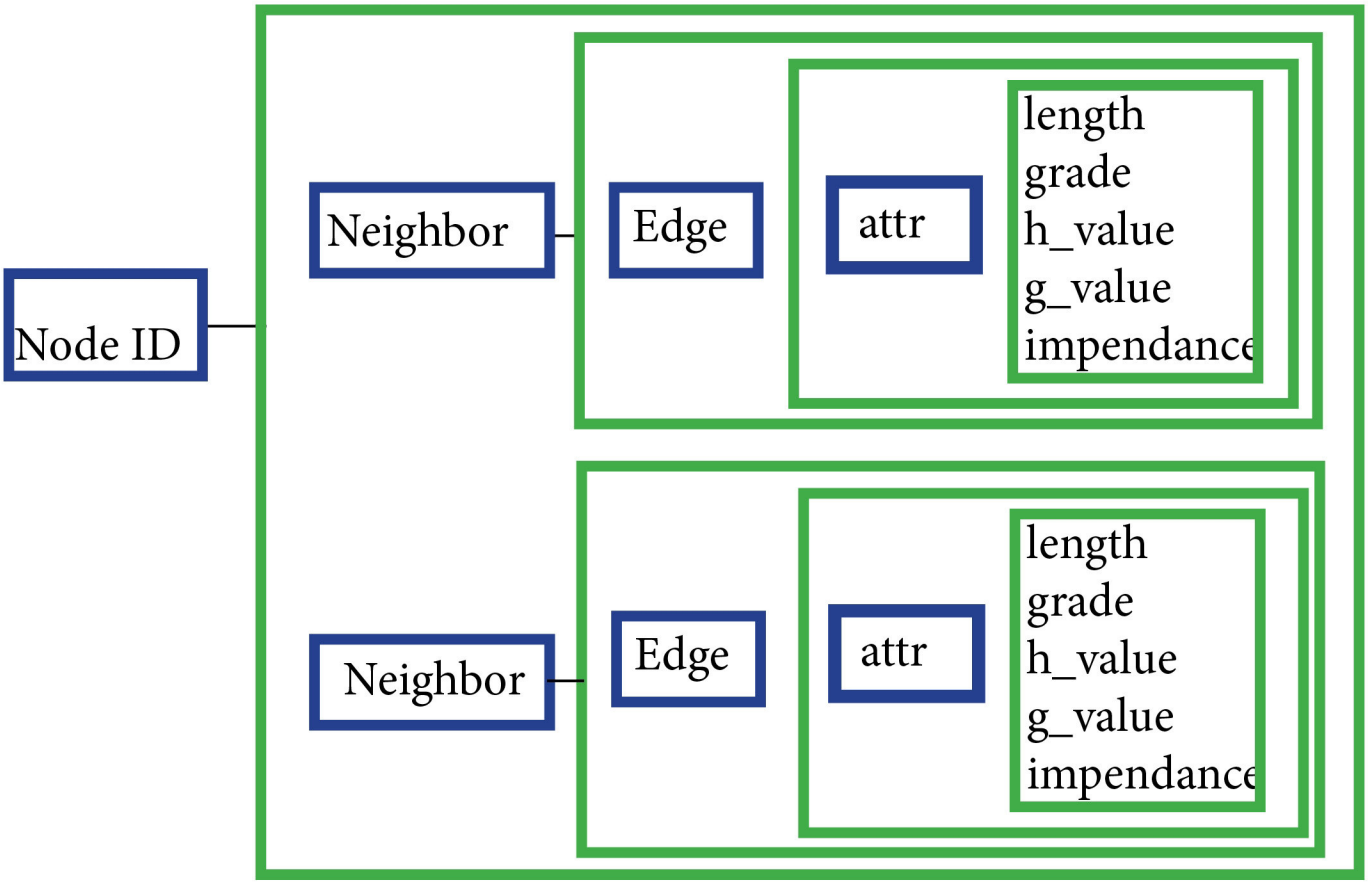


Advantage:

- Represents sparse street network that are usually directed
- Allows flexibility for node and edge insertion
- Allows instant access to value through key: O(1) on average
- Storage: O(V,G)

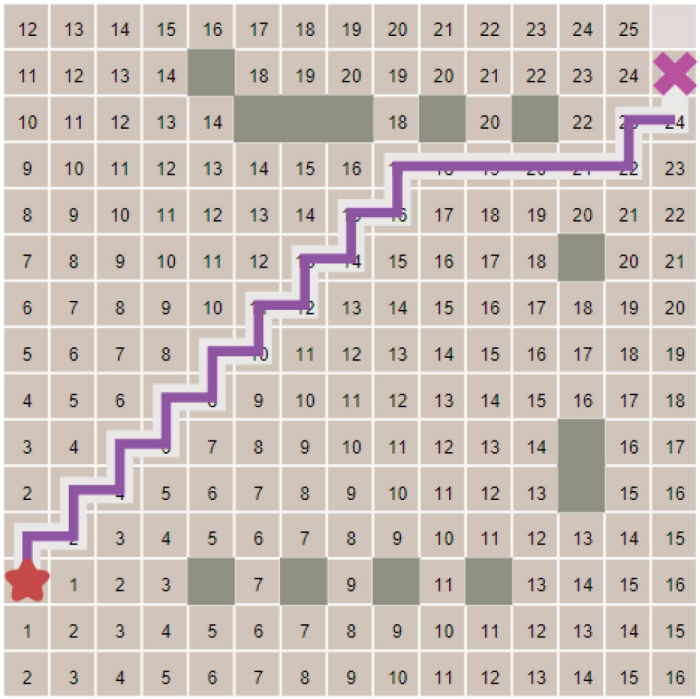
DATA DIAGRAM





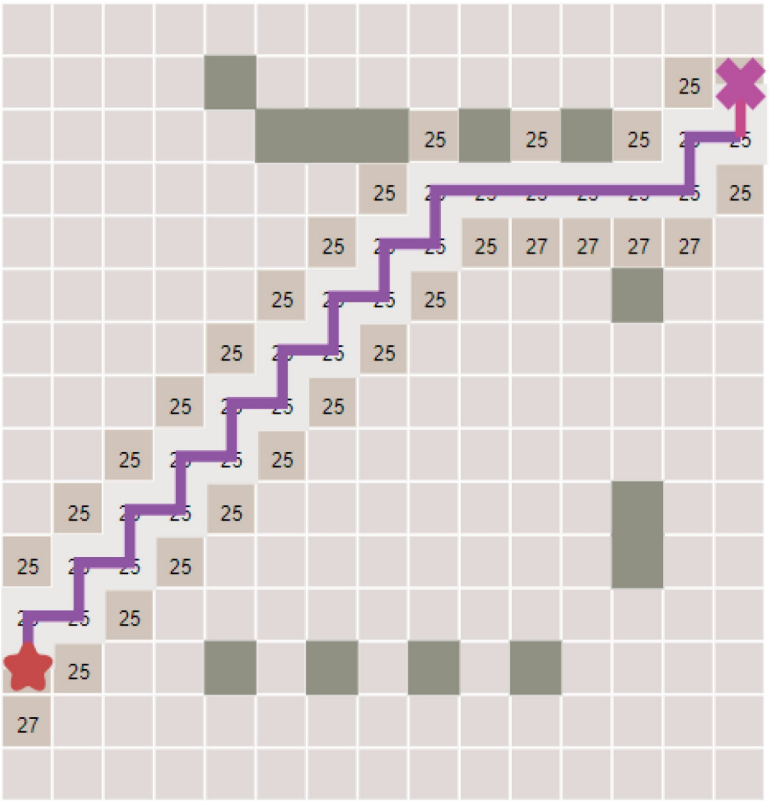
SEARCH ALGORITHMS

Dijkstra's



Dijkstra's algorithm from Osmx allows us to keep track of cost of the move for the past nodes and the current move being considered. We manipulated the weight function taking in length (distance between two nodes) and grade (slopes between two node's elevation) in order to minize the elevation within certain length.

A* Search



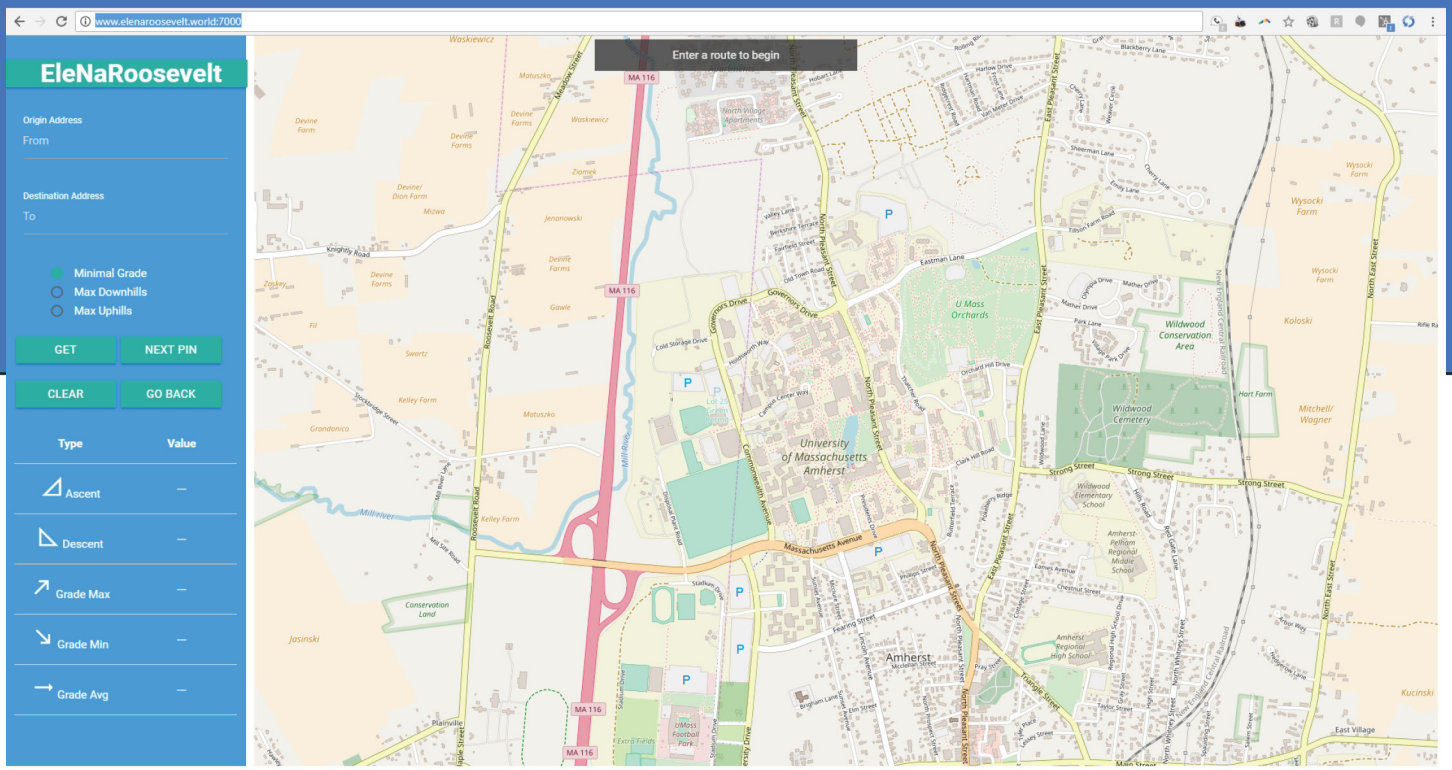
- Implemented Astar search to reduce the cost of searching for the shortest past from O(E log(V)) to O(E).

- Future work: Implement a boundary of search into the algorithm in order to maximize elevation gain within certain distance

USER INTERFACE

For interactive website for the elevation based navigation please visit

http://www.elenaroosevelt.world:7000/



- Enter address of origin and destination
- Choose the type of route to minimize grade, maximize downhills or uphills
- Click Get to get the route
- Click next pin to start navigation
- Click clear to start finding new route
- Click go back to go back to the previous spot

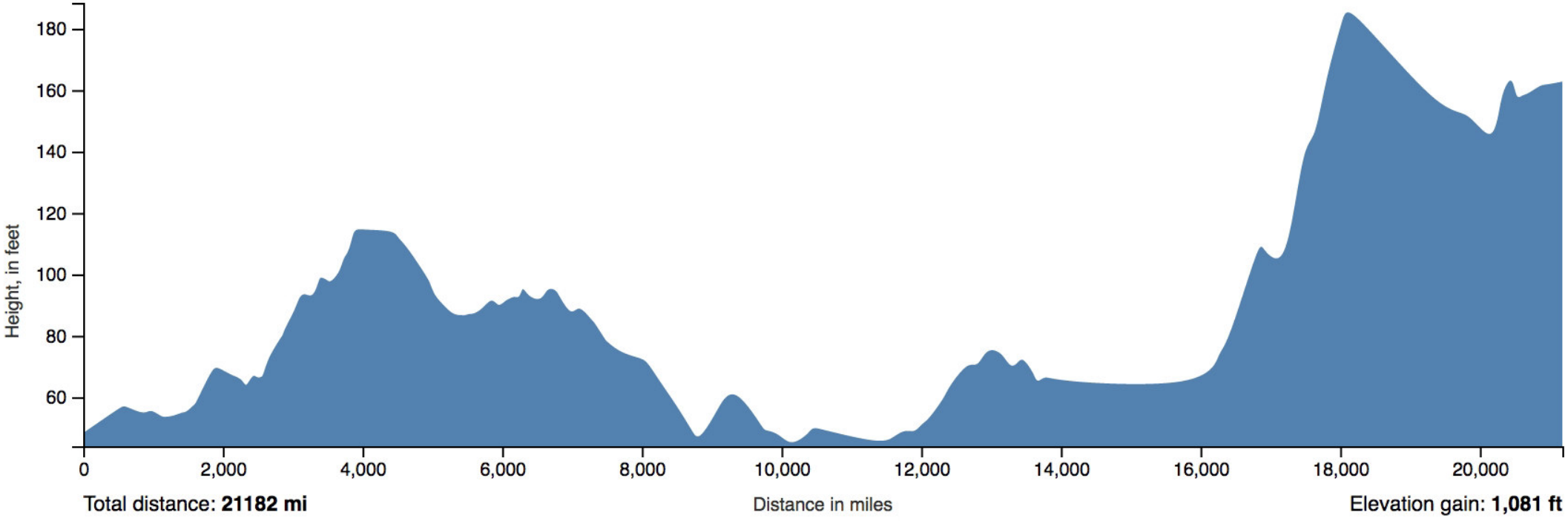
Origin

Destination

Minimize Slope

Follow Route

Route Length:
Shortest length:
Avg Grade:
Ascent length:
Descent length:


180
160
140
120
100
80
60
Height, in feet
0
2,000
4,000
6,000
8,000
10,000
12,000
14,000
16,000
18,000
20,000
Total distance: 21182 mi
Distance in miles
Elevation gain: 1,081 ft

CHALLENGES

Algorithm
- Come up with a weight function to minimize the elevation within certain distance
- Implement our own A* search algorithm to have more flexibility in manipulate the search options.
- Find an algorithm to maximize the elevation gain while keeping it at a certain distance

System
- Download large amount of street network takes time and large storage capacity.

User Interface:
- Decide API to work with: Google API only allows plotting 20 waypoints, Open Street Map allows more waypoints but having problem in displaying the route properly.

Time constrain:
- Many challenges can be solved if more times allowed. However we learned to make decisions and worked within a time frame to deploy an accessible working prototype to the Internet.

THE TEAM

Naman Shenoy nshenoy@umass.edu

the bitcoin master

Corey Collins cwcollins@umass.edu

the more artsy one

Jaeseong Lee jaeseonglee@umass.edu

the milk coffee life-saver

Akash Mantry amantry@umass.edu

the data ninja

Ly Harriet Bui bui23L@mtholyoke.edu

the one nd only Elena