**CS520 Final Project**

Yu-Chuan Chen, Huan Wang, Chenmei Li, Ruei-Yao Sun

**Introduction**

Nowadays, many navigation systems, such as Google map, can help us to choose a shortest or fastest route. But all these maps do not consider elevation gain. For example, if people who are hiking or biking, they may want to find a route that maximizes or minimizes elevation gain. Therefore, we design a system which user can choose two locations as input and then select whether to maximize or minimize. Moreover, user can define a percentage that compares the distance of route considering elevation gain to the path without consider it. In other words, if somebody want to have more uphill, they can select the maximize choice. Our group will focus on both sophisticated UI and effective algorithm.

**System Design**

* **Structure**

一張含有 文字 的圖片

自動產生的描述

Fig 1. System Structure

We will build a web interface where users can set their criterias including starting point, destination, and elevation gain they want. And this interface will send the query to our server, and the server will calculate corresponding path. For finding a certain path, it is straightforward to think of using some graph-based algorithm, so in the server, we use Python *openstreetmap* library to transform an area into a network graph and utilize *Networkx* library to develop our core algorithm to find that suitable path. After we get the path, we will display the route through *Google map api* to users.

For technical details, our interface is based on common front-end technologies such as HTML/CSS/Javascript, and the web framework is python *django*.

* **The design of User Interface**

Aside from HTML/JS, we also applied Bootstrap 4 and Jquery to build our UI. Firstly, we determined the style of our website, and selected the overall color scheme of the web page through the advice online. Then, according to the requirements, we designed the following website.

* + **Before you enter the locations:**

图片包含 文字

描述已自动生成

Fig 2. UI before search

1. It is our website logo, including the icon and name of our map. We designed this logo to match our entire web style. In particular, we used CSS to make the earth rotate.
2. In this section, you can enter the source and destination and here we can auto-complete the location you want to enter.
3. Click on this button, you can exchange the start point and destination without typing the location again. In this way, you can easily plan the round trip.
4. You can choose to minimize the elevation gain and you may go the extra mile to save you a couple thousand feet in elevation gain. Or you may want to maximize elevation gain if you are looking for an intense workout.
5. Here is the slider which you can select the percentage of shortest distance. We will generate a route that limit the total distance between the two locations to this value.
6. When you make all the choice about, you can click on the ‘Let’s Go!’ button, the best route will be displayed on the map on the right.
7. When you click on this button, you can enlarge the map to full screen.
8. Clicking on ‘+’ will zoom in the map, while ‘-’ will zoom out the map.

**After you submit your requirement:**

**一張含有 文字, 地圖 的圖片

自動產生的描述**

Fig 3. UI with result

1. We can see the best route in the map, and it will generate a card to show the best route’s statistics, including the total distance and elevation gain. Moreover, if you click on the ‘Show Details’ button, it will show you the directions in the top right of the map (8th section).
2. In this section, the detailed route will be displayed to you, and if you want to hide it, just click on the ‘Show Details’ button again.

* **Core algorithm**

Below is the pseudo-code of minimum elevation gain algorithm, and we can compute maximum elevation gain in a similar way. The essence of algorithm is Dijkstra’s algorithm, for more information, see min\_route.py and max\_route on GitHub.



**Evaluation**

We compare route provided by Google map with our min/max elevation gain algorithm. The source and destination are set to Umass Amherst Center (42.386918, -72.530459) to Amherst College (42.370966, -72.517142). And the percentage of shortest path for min/max elevation gain are set to 200%.



Google Map



Min elevation gain



Max elevation gain

Fig 4. Route Comparison

We show the elevation analysis by Google Map in Fig.4. The “Google Map” means the suggestion route for bicycling provided by Google Map. And the others indicate our algorithm’s results. Since Google Map has limited the number of waypoints, (node in a path), when plotting a route, we only take every n nodes in our algorithm’s found path to plot. (n=Number of node in our algorithm’s path/Upper Limit of Google Map) In other words, the above plots do not fully reflect to our algorithm’s true performance. However, considering such deviation, we still can find out that the “Max elevation gain” route is more steep that “Google Map”, while the “Min elevation gain” is flatter. Therefore, we can claim that our algorithm do find useful paths.

**Supplementary Material**

Github Code: <https://github.com/msps9341012/Elena-Project>

Demo Video:

<https://drive.google.com/file/d/14tDJSehiU-h9s8sZCU7ZGvzoimRFzBNU/view>