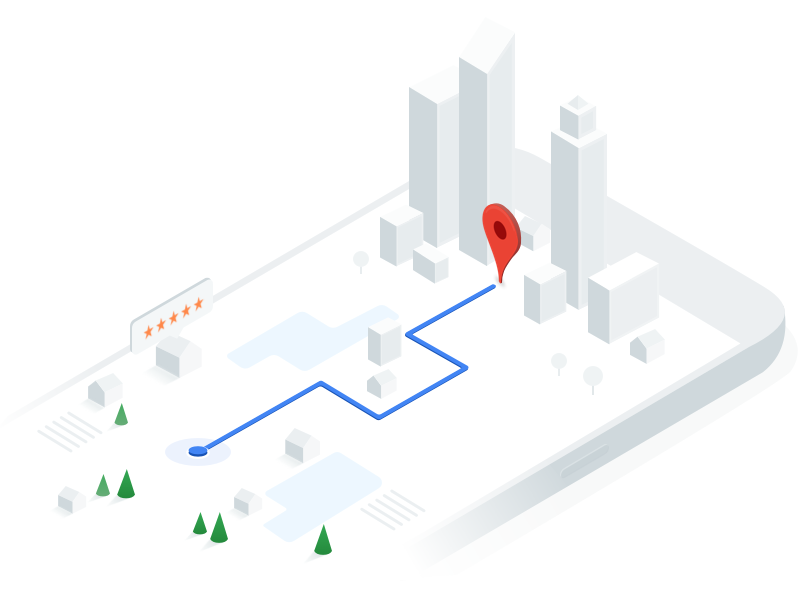
JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

NOIDA

A Case Study of Google Maps





Dr. Manish Kumar Thakur Sunny Dhama (17103071)

Siddharth Batra (17103070)

Parth Chandna (17103076)

Rahul Kumar (17103075)

Introduction

The aim of our project is to study and implement the algorithms used in the daily maps and navigation providing services. Mainly targeting Google Maps for our study. We will study how the service works, the algorithms it uses, how they are beneficial from other algorithms already present and try to implement the basics of its.

Google Maps

Google Maps is a web mapping service developed by Google. It offers satellite imagery, aerial photography, street maps, 360° panoramic views of streets (Street View), real-time traffic conditions (Google Traffic), and route planning for traveling by foot, car, bicycle and air (in beta), or public transportation.

Google Maps began as a C++ desktop program at Where 2 Technologies. In October 2004, the company was acquired by Google, which converted it into a web application. After additional acquisitions of a geospatial data visualization company and a real time traffic analyser, Google Maps was launched in February 2005. The service's front end utilizes JavaScript, XML, and Ajax. Google Maps offers an API that allows maps to be embedded on third-party websites, and offers a locator for urban businesses and other organizations in numerous countries around the world. Google Map Maker allowed users to collaboratively expand and update the service's mapping worldwide but was discontinued from March 2017. However, crowd sourced contributions to Google Maps were not discontinued as the company announced those features will be transferred to the Google Local Guides program.

Google Maps' satellite view is a "top-down" or "birds eye" view; most of the high-resolution imagery of cities is aerial photography taken from aircraft flying at 800 to 1,500 feet (240 to 460 m), while most other imagery is from satellites. Much of the available satellite imagery is no more than three years old and is updated on a regular basis. Google Maps used a variant of the Mercator projection, and therefore cannot accurately show areas around the poles. However, in August 2018, the desktop version of Google Maps was updated to show a 3D globe.

How does Google Maps work?



Google Maps is one of the best inventions that have happened till date in terms of mapping and navigation, as we all rely on maps whether we need any information about the routes, traffic. It gives us near real time traffic updates and it uses our data such as path, distance and time to tell near real time traffic.

Google takes help from the following service providers and services to get the data and real time information:

• Map partners: Google has partnered with various data sources with the help of Base map partner program and a large number of agencies submit data to Google.

• Street view: Based on the GPS coordinates of vehicles Google cover the surface its street view images.

• Satellites: Google map also add satellite view with the collaboration with Google earth.

• Location services: Google access the location data collected by your smartphone.

• Google Map makers: it is a way which allows anyone to contribute their local knowledge top Google maps.

• Local Guides: As well as its army of editors, Google also has millions of so-called local guides. When you’re in Google Maps, go to My Contributions and you can search for different places in your area. By leaving a review, answering a few questions, and submitting a photo, you can contribute to this additional layer of data.

To process all these data and to answer different queries from the user Google uses various algorithms to find a solution. These algorithms include but not limited to:

• Spatial indexing algorithms and algorithms of computational geometry to organize the map data and retrieve it efficiently.

• Algorithms to draw maps (e.g. project latitude and longitude coordinates, fill the polygons, place names for streets, cities, businesses, parks, etc.).

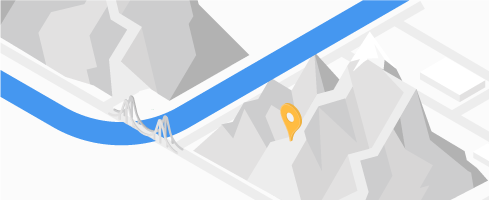
• Algorithms to understand queries from users (NLU or NLP)

• Algorithms to process GPS signals (on which I am much less expert than on other topics here)

• Algorithms to perform what is called geocoding, converting addresses to points (or polygons) on a map.

• Algorithms to perform reverse geocoding, converting points to addresses using point in polygon algorithms (another example of using computational geometry).

How does Google Maps find a path?



There are a number of complex aspects of maps and navigation especially those relevant to computing routes.

• Gathering and organising the data: The data about the longitudes and latitudes, complex geometry and map coordinates. The data about the roads, addresses, business, parks, malls, and institutions are needed.

• The most important thing is the cost (weights). Calculating these weights are a complex task altogether. Some of the complex paths of determining the costs are:

* + Collecting and using traffic data.
  + Time to travel through intersections
  + Amount of left and right turns in the route (for left hand drive countries and vice versa for right hand drive), and many more.



Google Maps also reply on real time mobile phone usage to calculate traffic conditions

What algorithms does Google Map use?

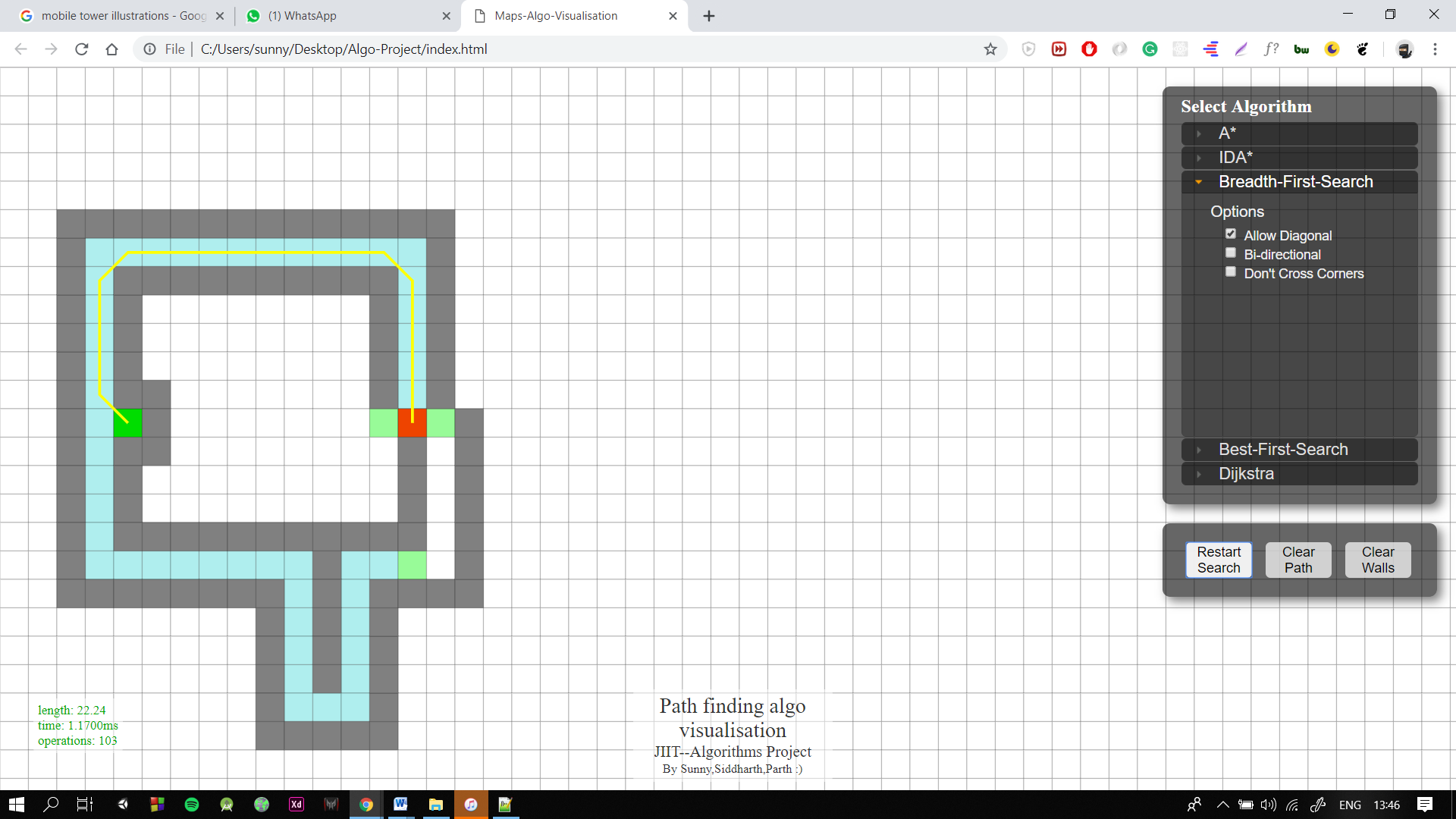
They are (but not limited to)

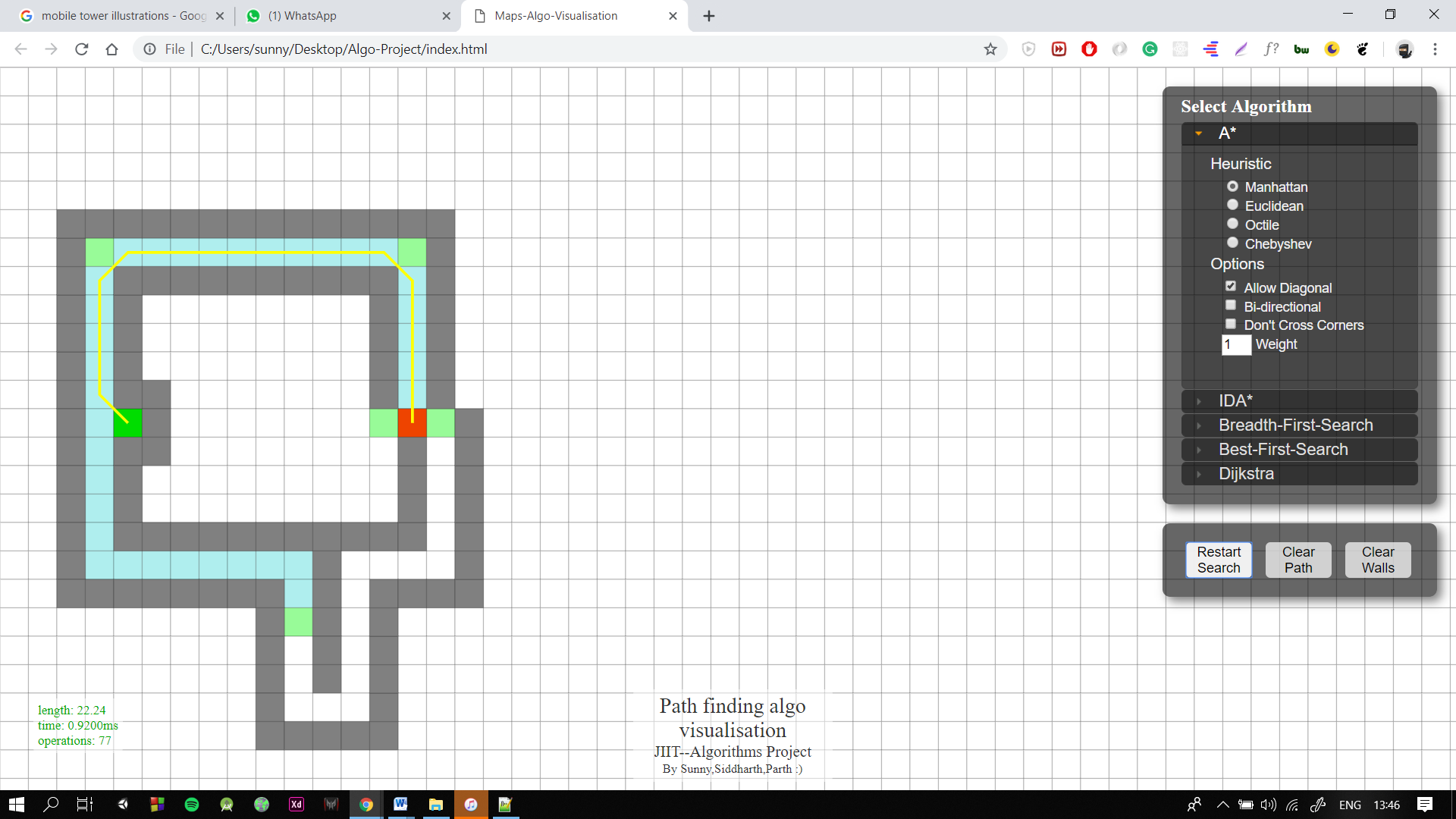
* Breadth-First-Search
* A\*
* Best-First-Search
* Dijkstra



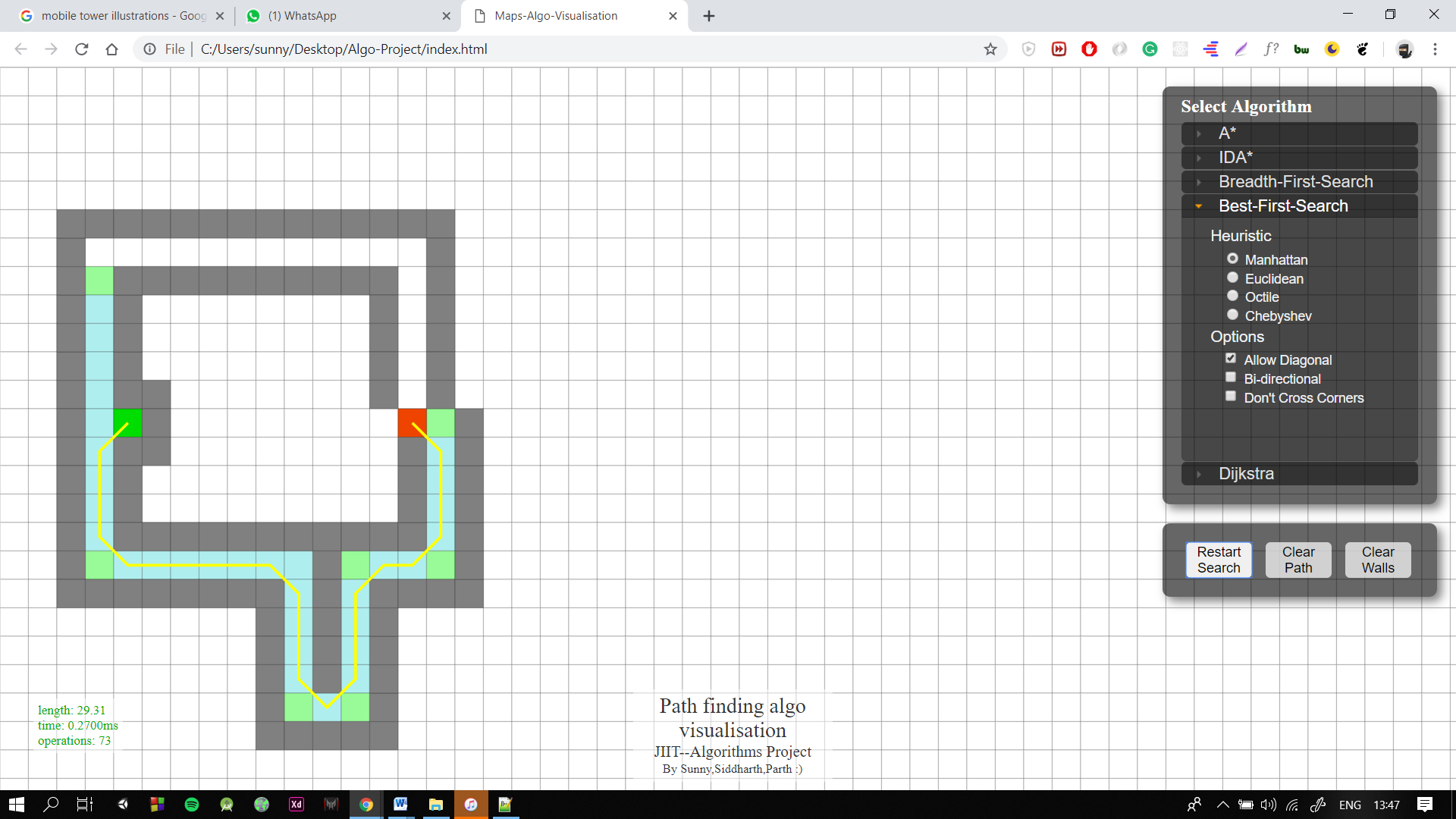
Which Algorithm fits best?

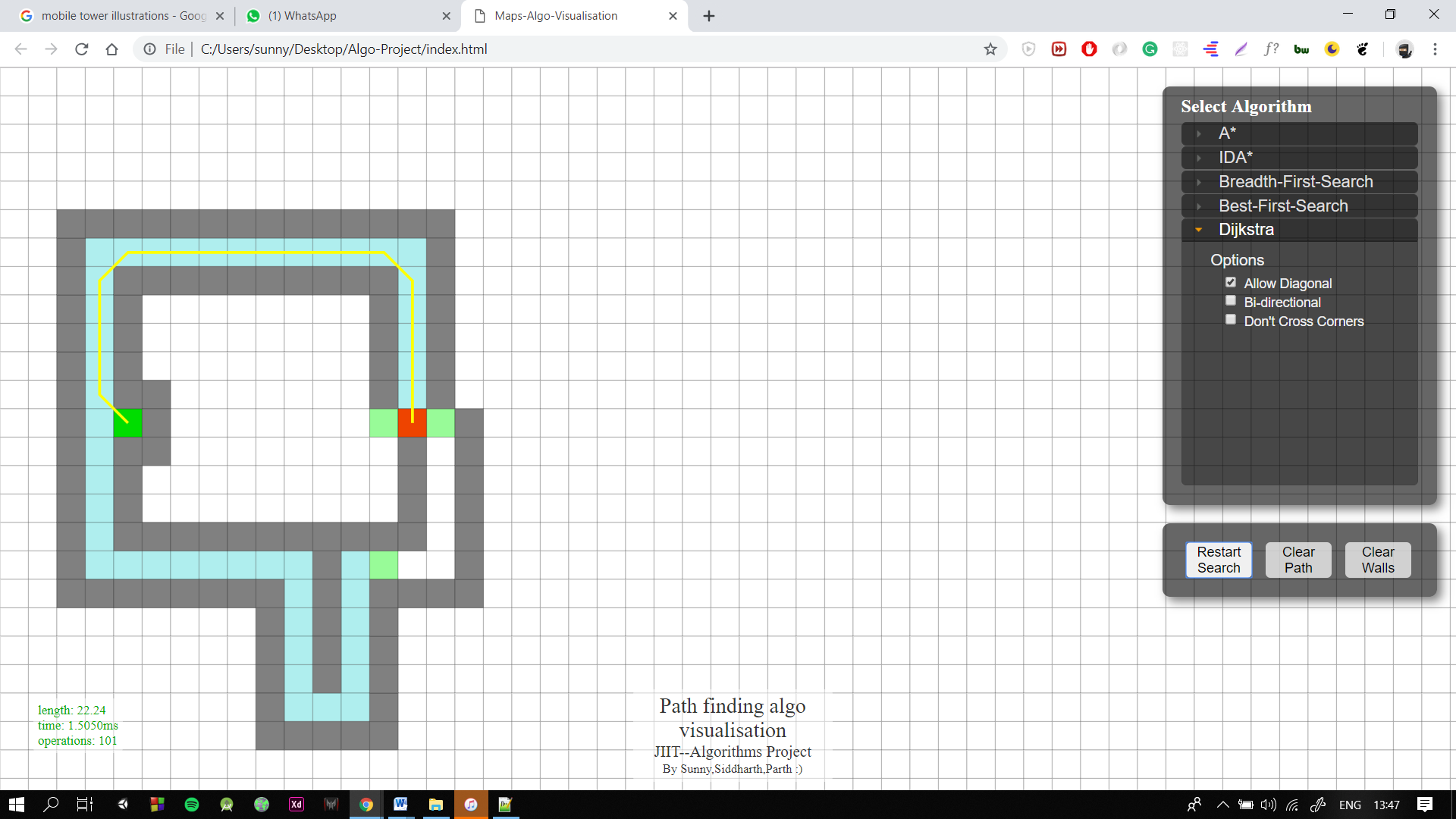
For this purpose we have developed an environment in which we have tested various algorithms visually and counted the number of recursive paths and iterations the program has gone through.

Breadth-First Search

A\* Search

Best-First-Search



Dijkstra

From this we have calculated the following points

* Length of path. (L)
* Time (T)
* Operations (O)

A mathematical equation with these variables (L, T, O) along with some constants may give us the required shortest path in shortest possible time.

In our research we also found some other factors that affect the path response, these are as follows

* Spatial indexing algorithms and algorithms of computational geometry to organize the map data and retrieve it efficiently.
* Algorithms to draw maps (e.g. project latitude and longitude coordinates, fill the polygons, place names for streets, cities, businesses, parks, …).
* Algorithms to understand queries from users (NLU or NLP)
* Algorithms to process GPS signals (on which I am much less expert than on other topics here)
* Algorithms to perform what is called geocoding, converting addresses to points (or polygons) on a map.
* Algorithms to perform reverse geocoding, converting points to addresses using point in polygon algorithms (another example of using computational geometry).

More of the other complex aspects of maps and navigation especially those relevant to computing routes.

* Gathering and organizing the data: it can come from many different sources. Just one example of a very complex problem that arises from that is the very messy geometry that results when the map coordinates from two sources (latitude, longitude) are in slight disagreement – especially polygons and lines. Besides data about roads, data about addresses, business, parks, malls, and institutions are needed.
* “Costs”. Yes that’s important and not at all trivial for various reasons. The costs are the “weights” on the graph’s edges to use the terminology of graph theory, but computing those weights accurately is complex. Here are some complex parts of determining costs:
  + - Collecting and using traffic data so that slow roads are less likely to be chosen in the route calculation.
    - Estimating future costs when route is long and is driven in the future.
    - Time to travel through intersections of roads. All navigation products take into account the differences in time needed to make different turns at an intersection because the times can vary significantly – left turns usually take much longer than right turns but not always. And the many and complex varieties of configurations of intersections make this a challenging problem.
    - Additional parameters to consider are fuel costs and the complexity of the route. If a route with a few turns is only slightly longer than a route with many turns, it is likely to be a better choice for the driver.

Route calculation algorithms themselves but remember it’s only one component of many in the map and navigation technology. As some have mentioned computing a route a long way across a large map can be expensive and slow. So *optimizations of the Dijkstra are needed*.

* A\* has been a very important concept in traditional AI, and many route calculation algorithms include it. But, in my experience in developing the algorithms, A\* has offers only a minor improvement in performance – maybe 30%. For other applications in AI, the graphs might be very different in such a way the A\* provides a larger performance advantage, but not much for road networks.
* Bi-directional. This is a somewhat more effective optimization and almost all route calculation algorithms in the industry use it but not primarily for the performance advantage it brings. It means that the route is computed both forward from the origin (with Dijkstra, A\*, reach-based routing, highway hierarchies, contraction hierarchies, or other methods) and backward from the destination. Like A\*, the technique reduces the area searched, but more importantly, some more important algorithms (ones using road hierarchies) require it to work. But bi-directional route computations bring their own complications – the algorithm must determine when the two searches have met at a point on an optimum or near optimum route and that’s more complex than it sounds.
* Using the road network hierarchy. This is **the most important performance optimization** and was used long ago at companies like Etak in a practical but not academically formal way. My invention and paper on reach-based routing brought the concept into academia and spawned further research producing algorithms like Highway Hierarchies and Contraction Hierarchies. Since the lead author of the papers on Highway Hierarchies, Peter Sanders, gave a talk at Google, it has been speculated that Google has used Highway Hierarchies:
* Shortcuts. This concept replaces sequences of edges with single edges in the preparation of the graph. The concept was used in industry before I published my paper and I regret not having space to mention it under Further Work in my paper. The folks at Karlsruhe in Germany took full advantage of shortcuts in Highway Hierarchies and Contraction Hierarchies.
* Other issues affecting the algorithms:
  + - Even if we ignore performance, Dijkstra does not solve all problems. One problem is time-dependency. Sometimes the algorithms must allow the costs on the edges to change depending on the time that the route brings the user to particular point on the roads.

For **example**, the road might be closed at a particular time.

* + - Or traffic might be predicted to become worse.
    - Most of the fast algorithms require processing on the map data before any routes are computed, but some of the data affecting that “pre-processing” changes quickly, e.g. traffic data. Determining what part of the problem is solved in pre-processing and what part is solved during the route calculation and how it is solved is very difficult.