



# VOYAGER MAP

Oral Presentation 2 by cd-087

Gengyang Wang

Yanchi Fang

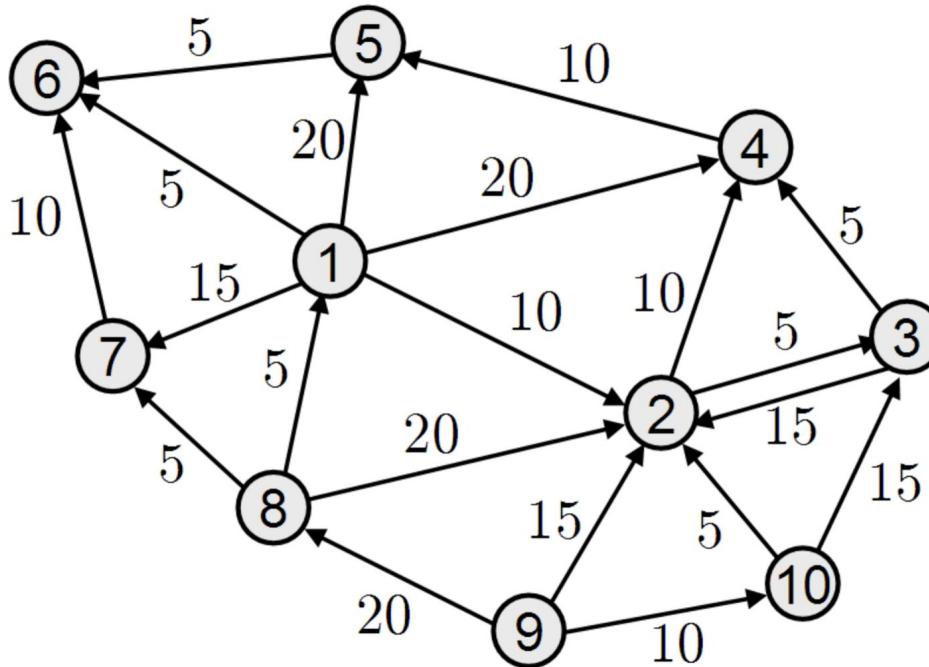
Xin Zhao

# BASIC FEATURES

Intersections, streets, features

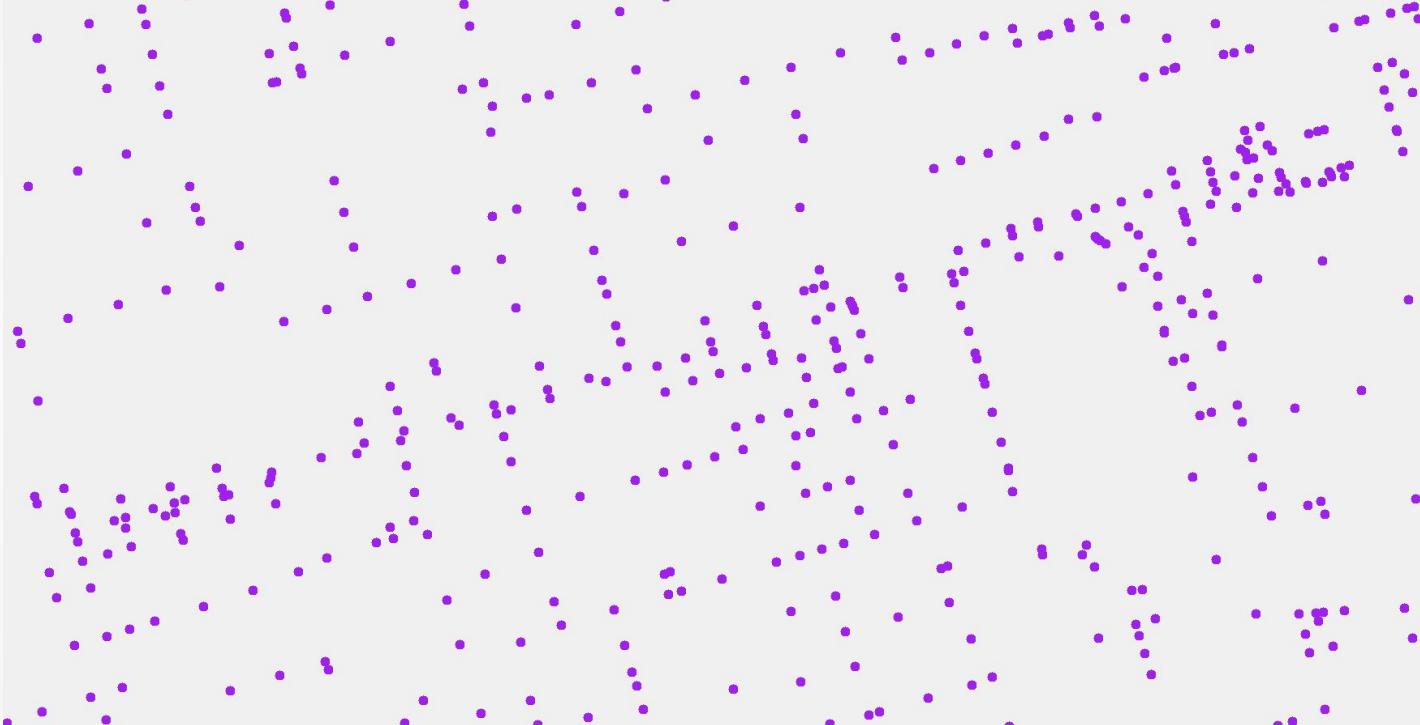
- Structure
- Streets & names
- Other Features

What do people normally think a map structure is?



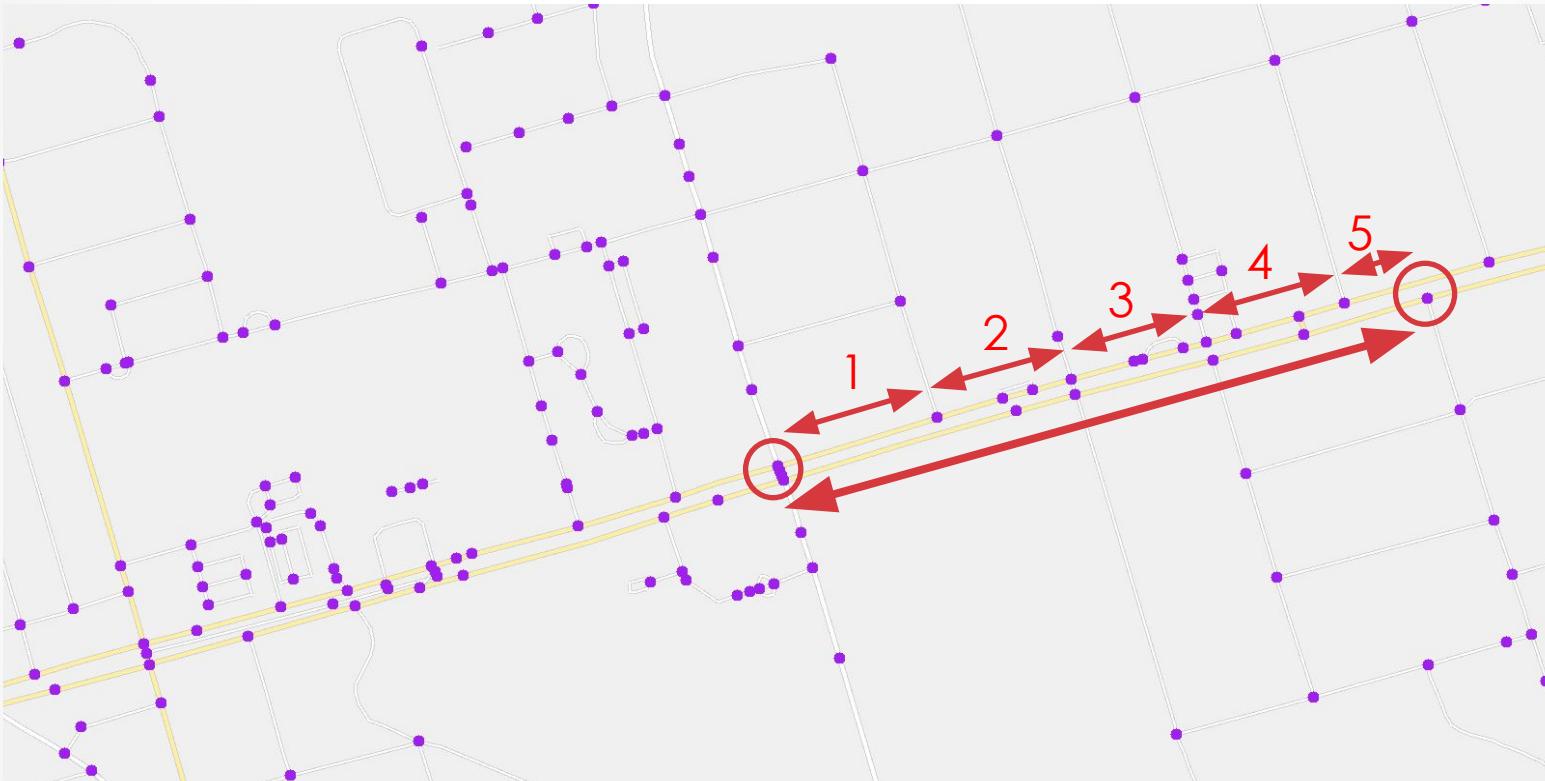
- People intend to think a map as graph structure

What do people normally think a map structure is?



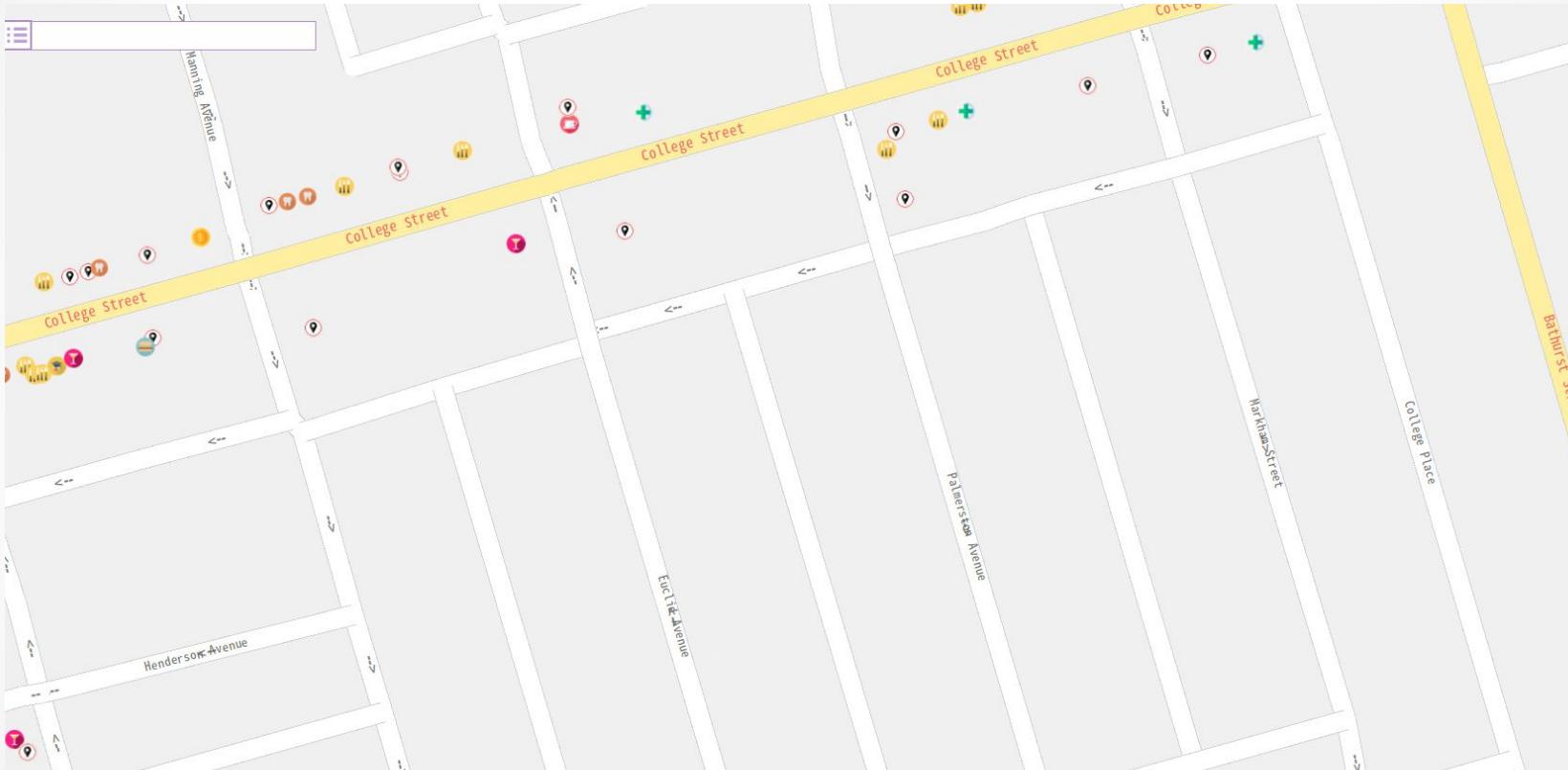
- A map only contains intersections

# What do people normally think a map structure is?



- Intersections joint by streets

# How are street names displayed?



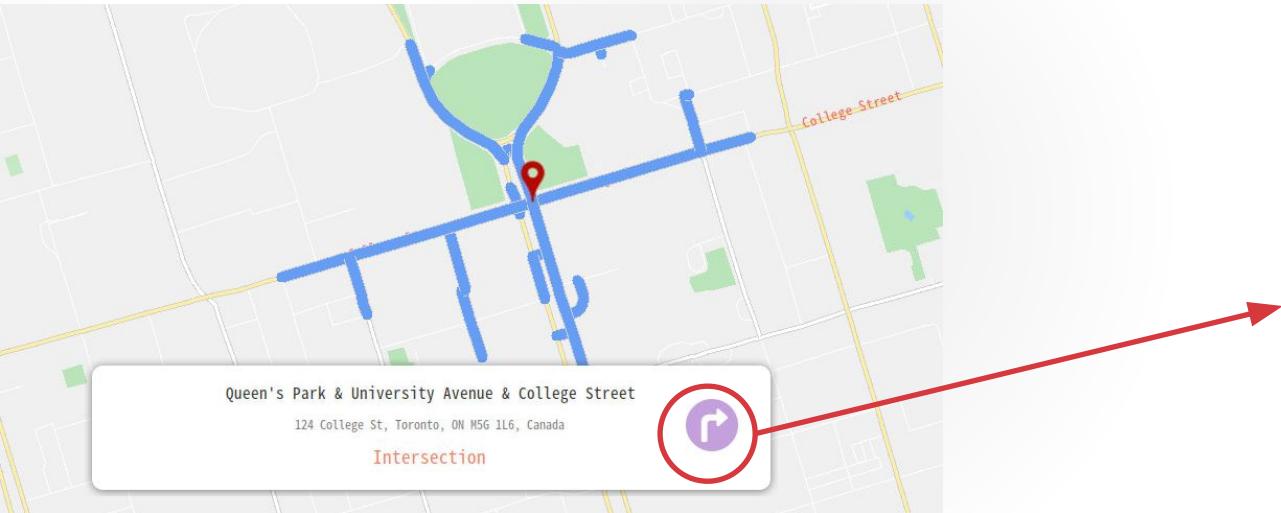
- Street names are aligned with streets

# How are street names displayed?



- Street names are aligned with streets

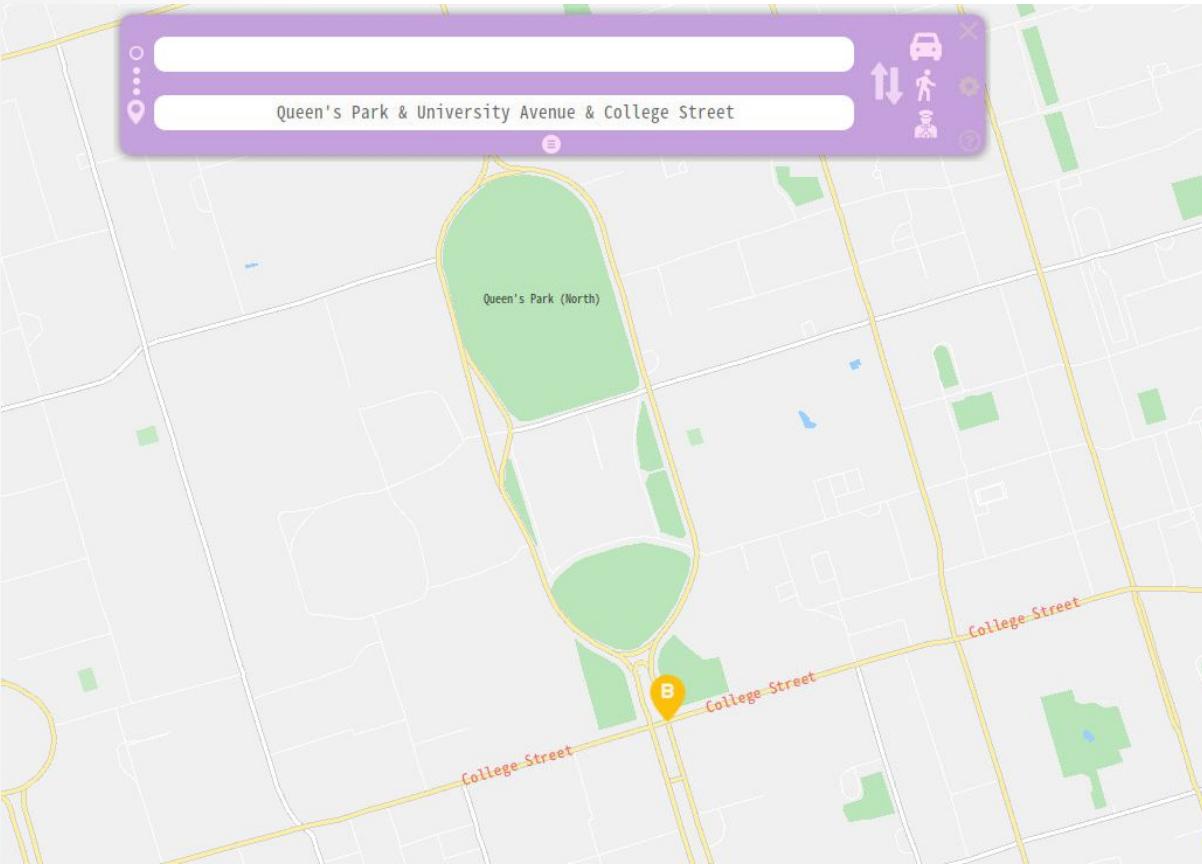
# What can intersections do?



- Street names are aligned with streets

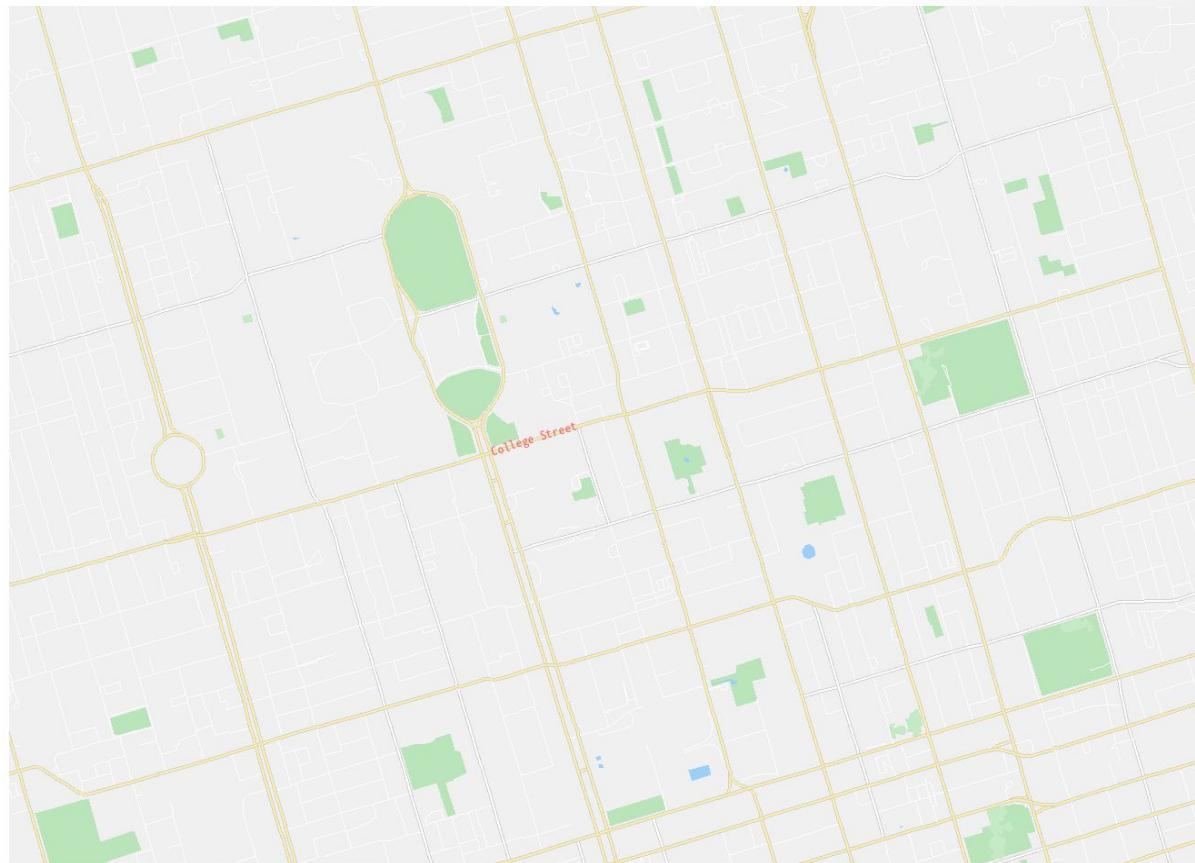
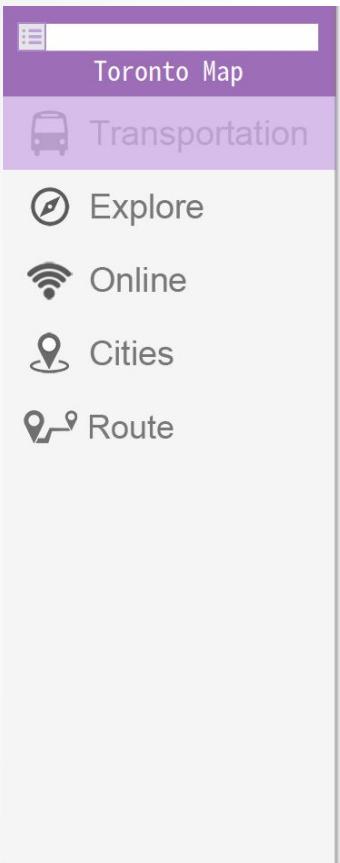
This button can automatically set this intersection as destination

# What can intersections do?



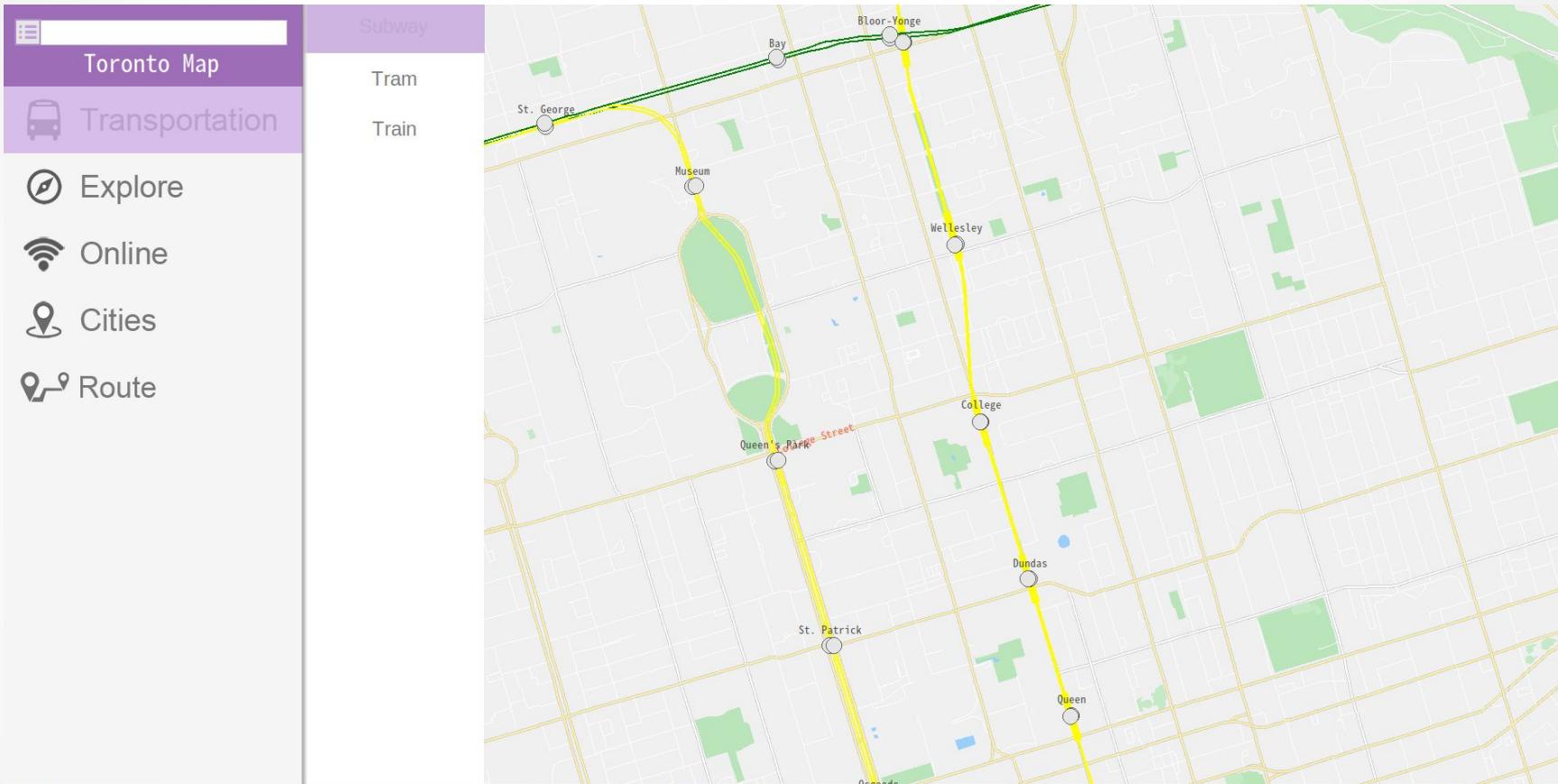
- Street names are aligned with streets

# How are street names displayed?



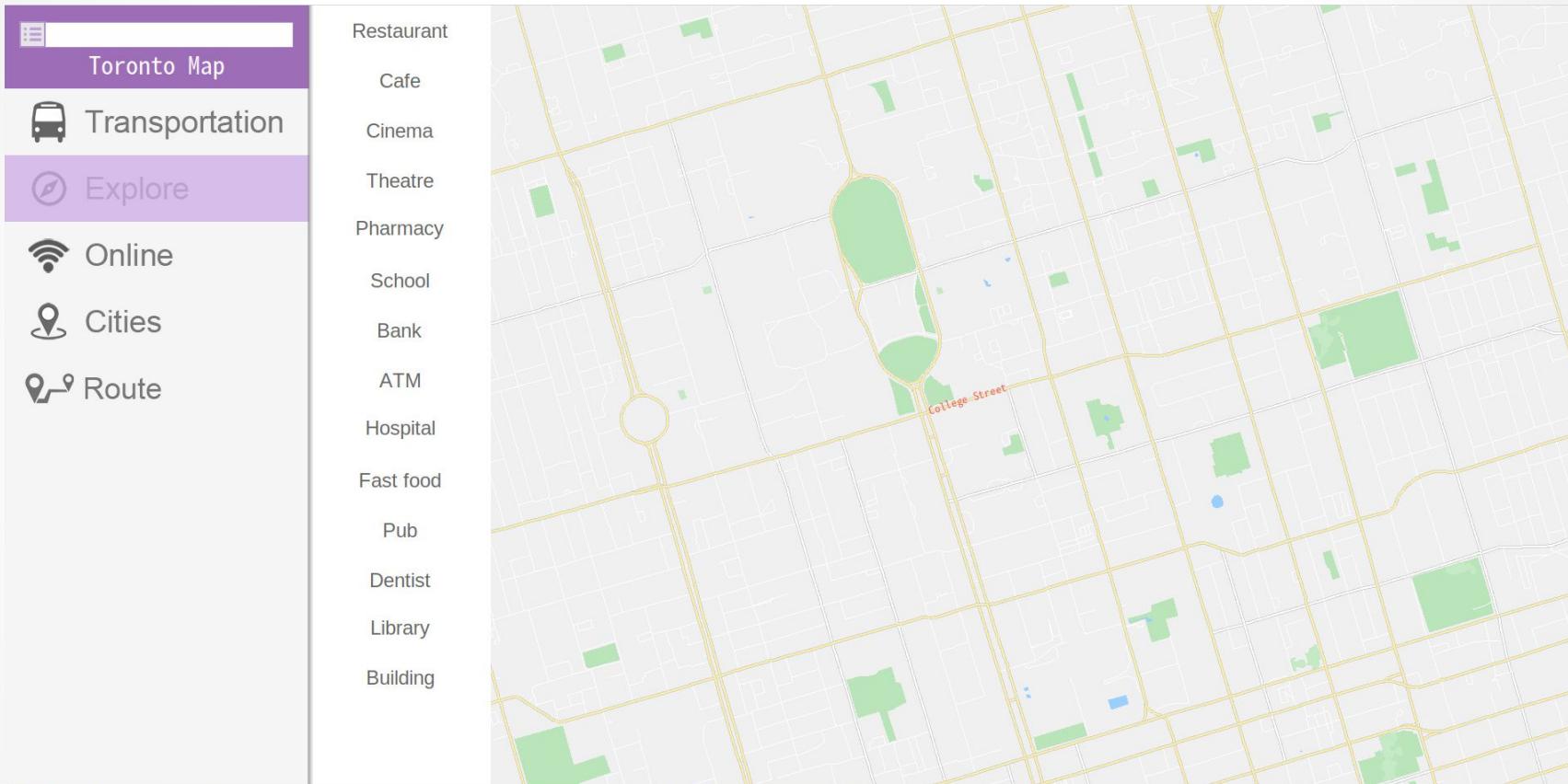
- Street names are aligned with streets

# How are street names displayed?



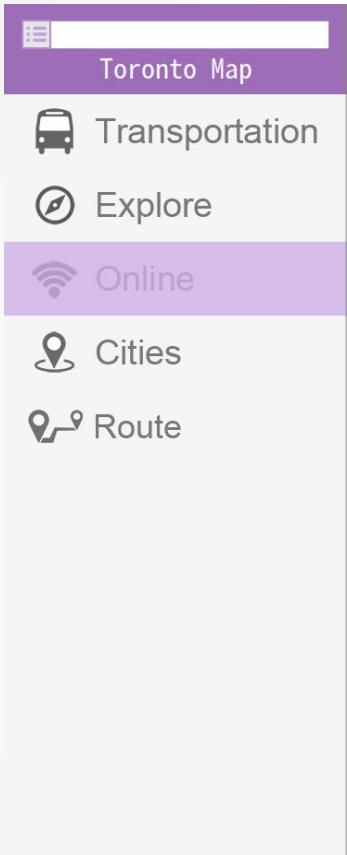
- Street names are aligned with streets

# How are street names displayed?

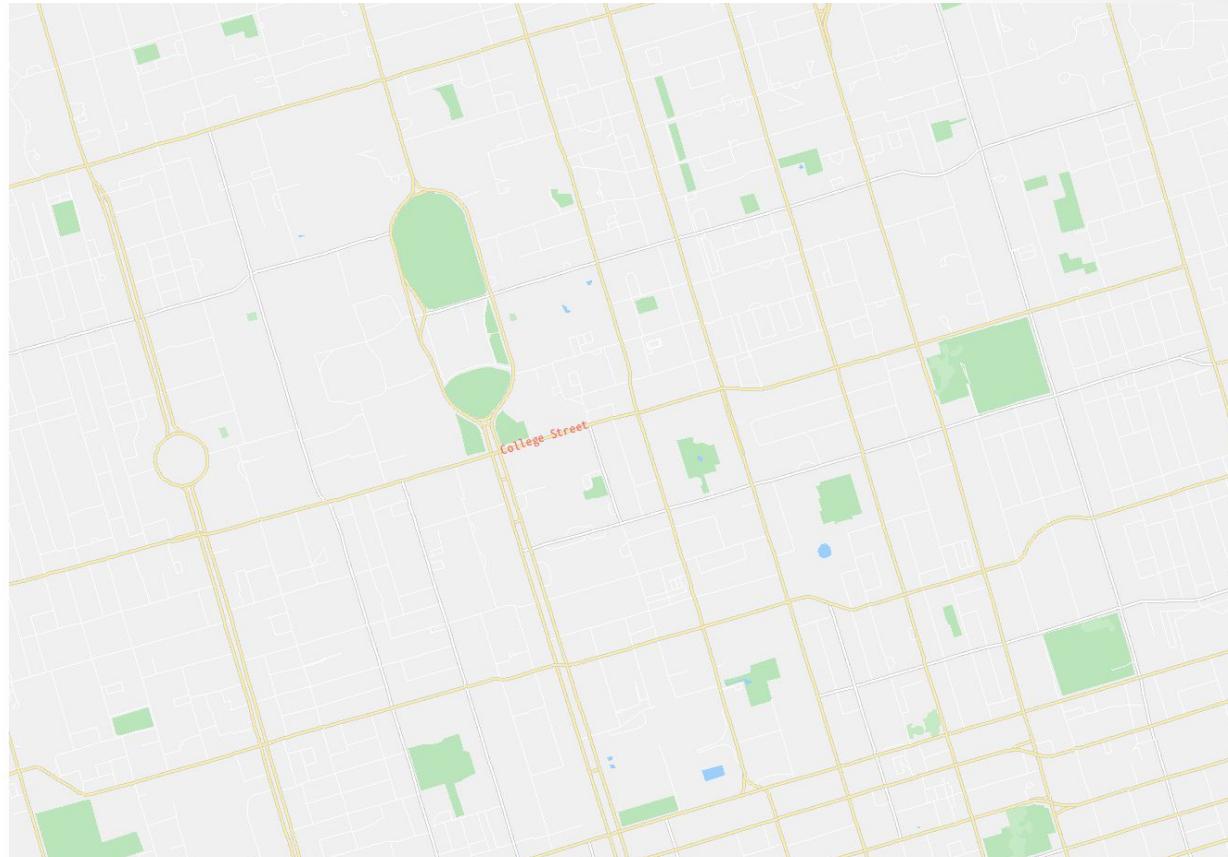


- Street names are aligned with streets

# How are street names displayed?

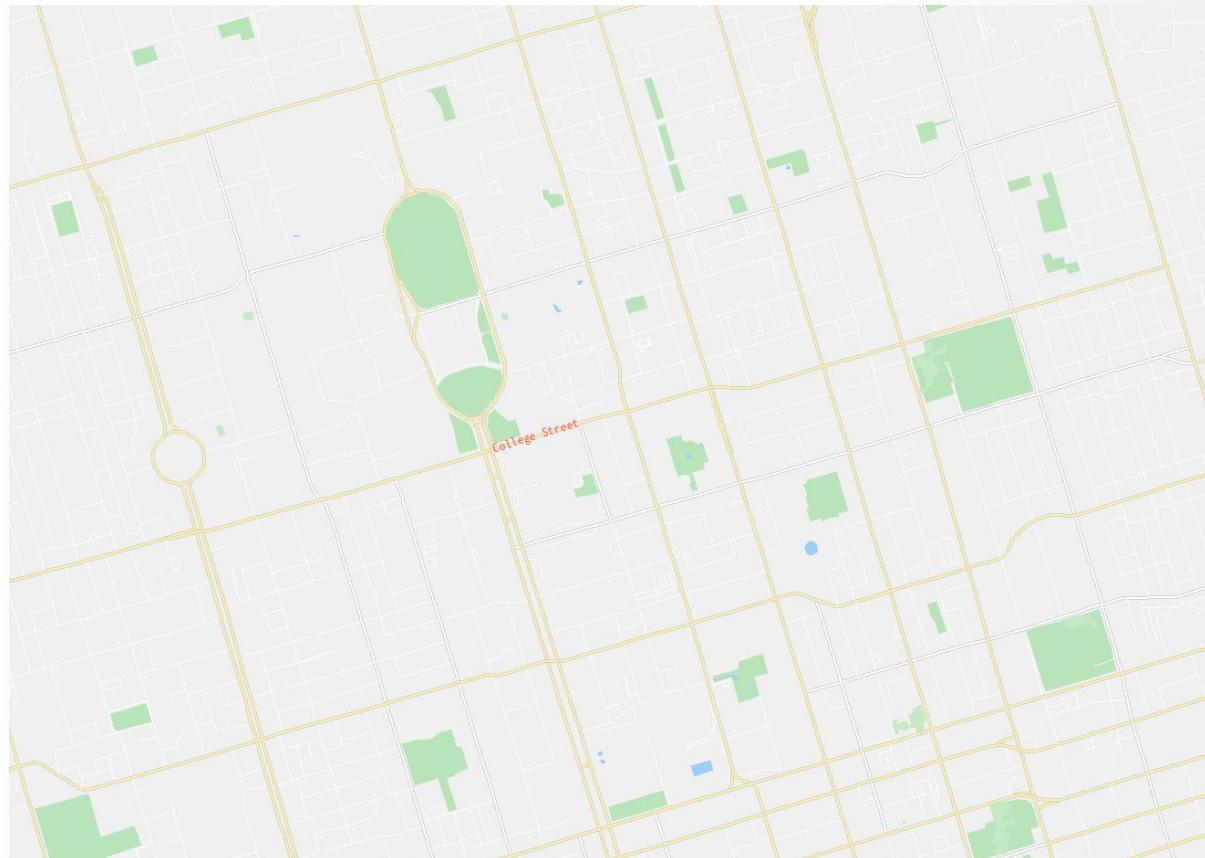
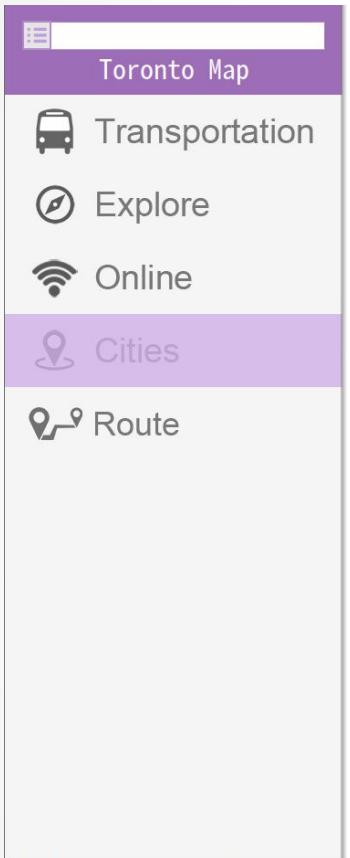


Weather  
Closure



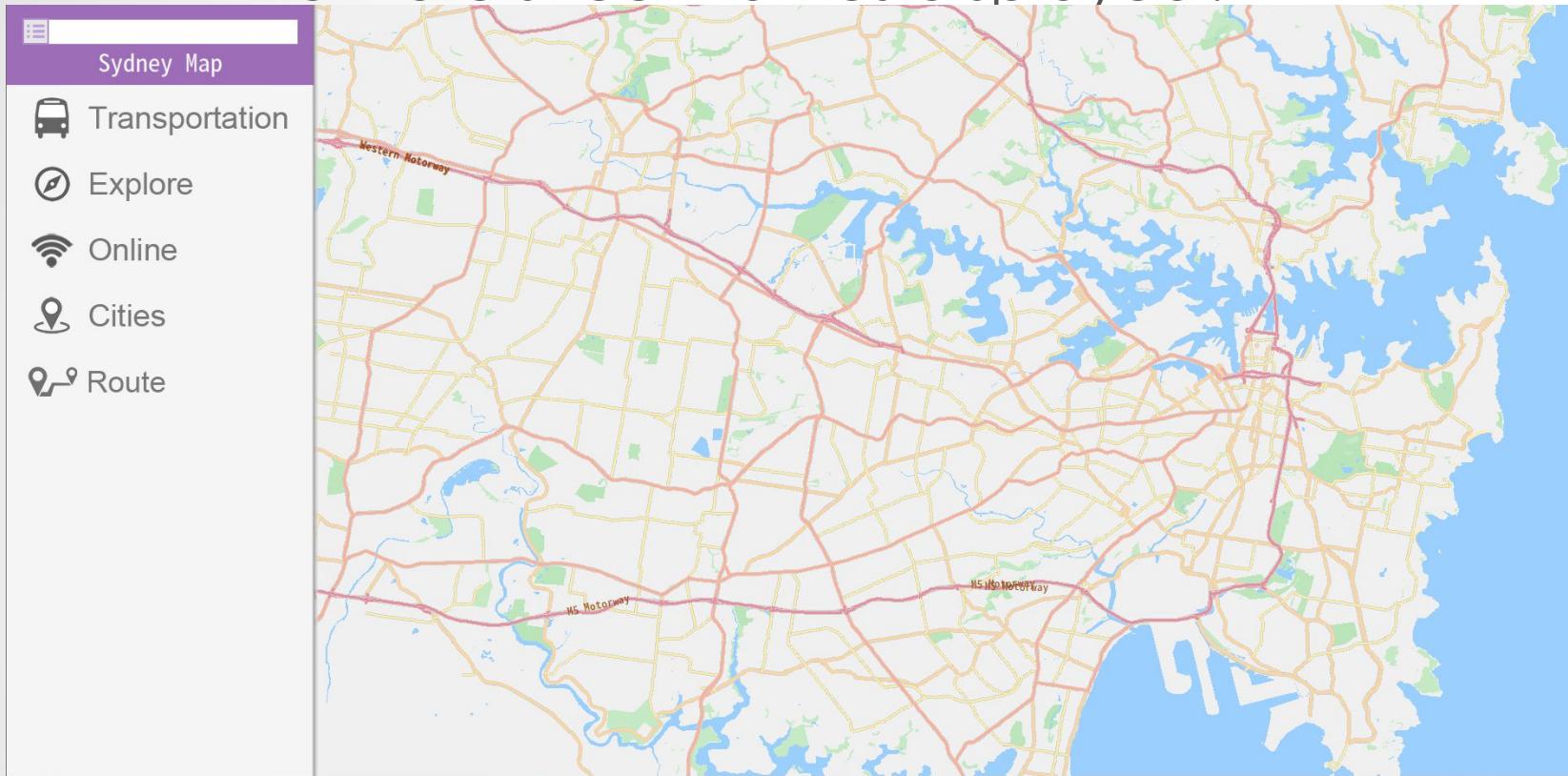
- Street names are aligned with streets

# How are street names displayed?



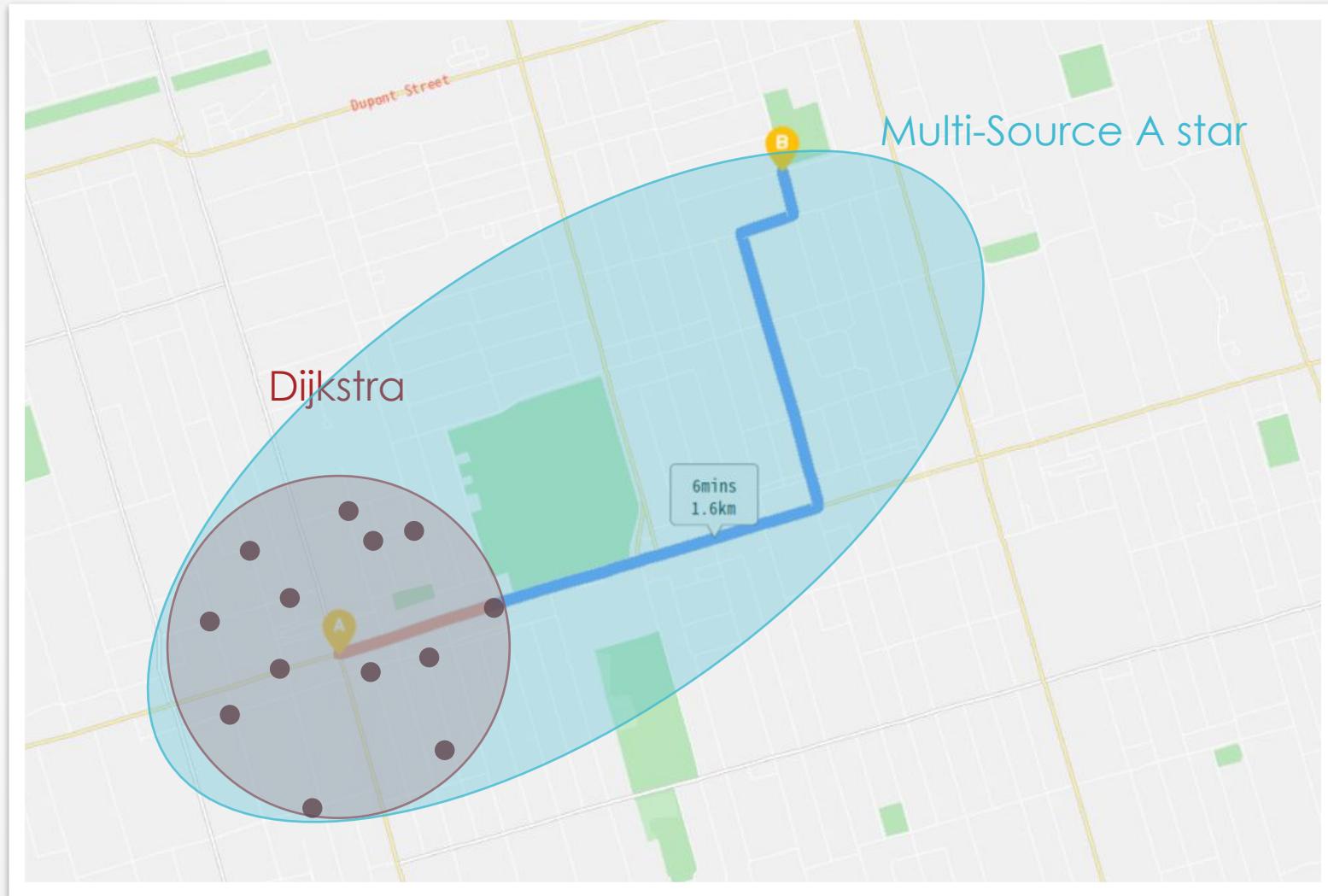
- Street names are aligned with streets

# How are street names displayed?



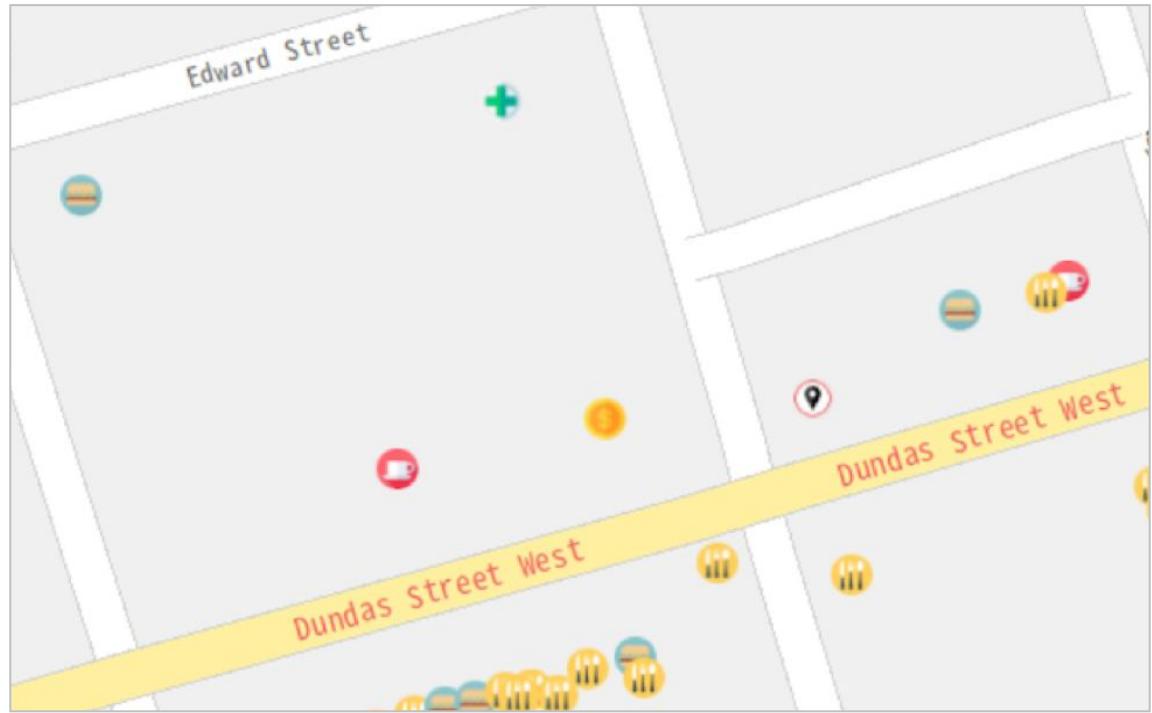
- Street names are aligned with streets

# Walk to Pickup Algorithm



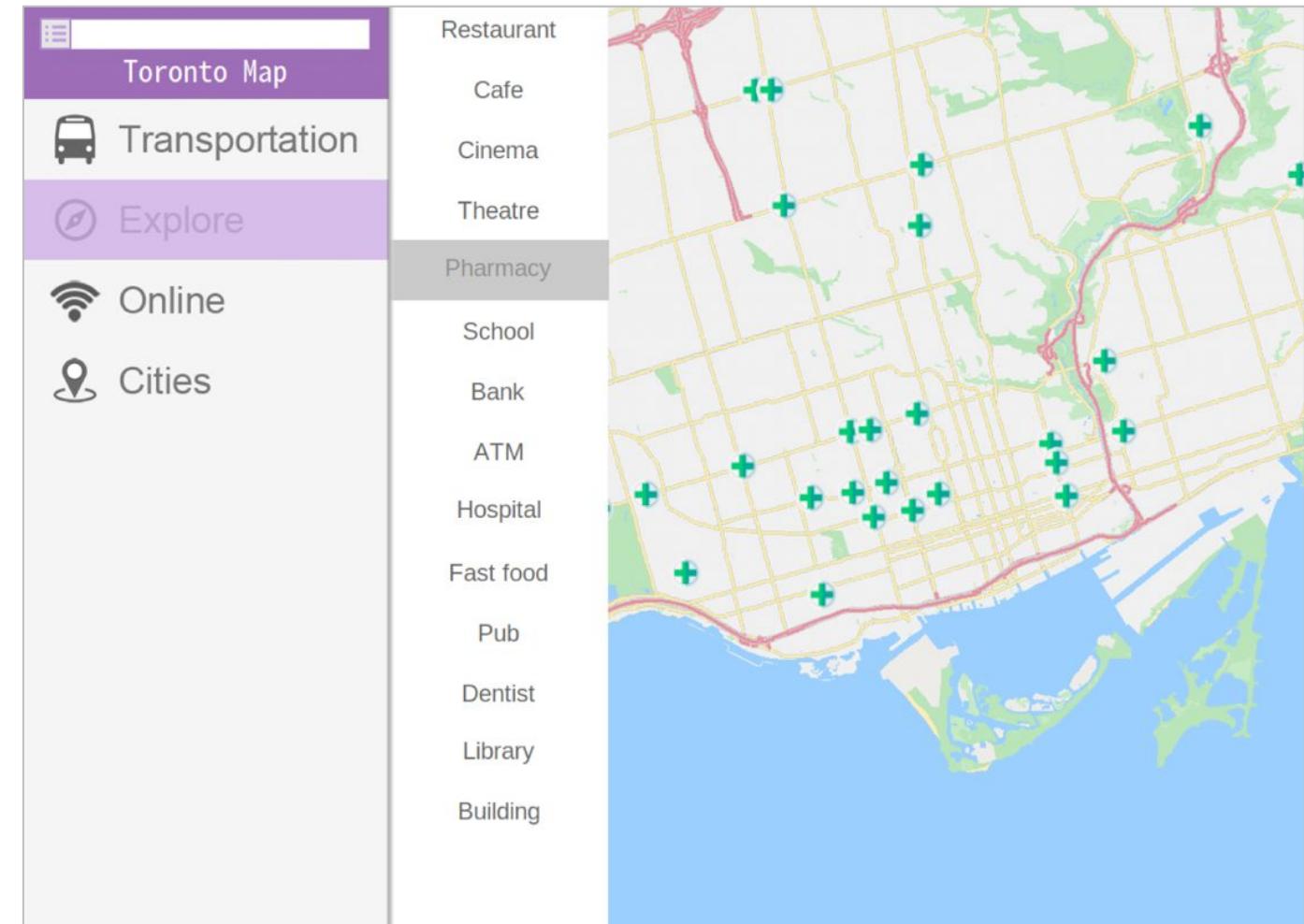
# POINT OF INTEREST (POI)

-  restaurant
-  cafe
-  cinema
-  theater
-  pharmacy
-  school
-  bank
-  ATM
-  hospital
-  fast food
-  pub
-  dentist
-  library



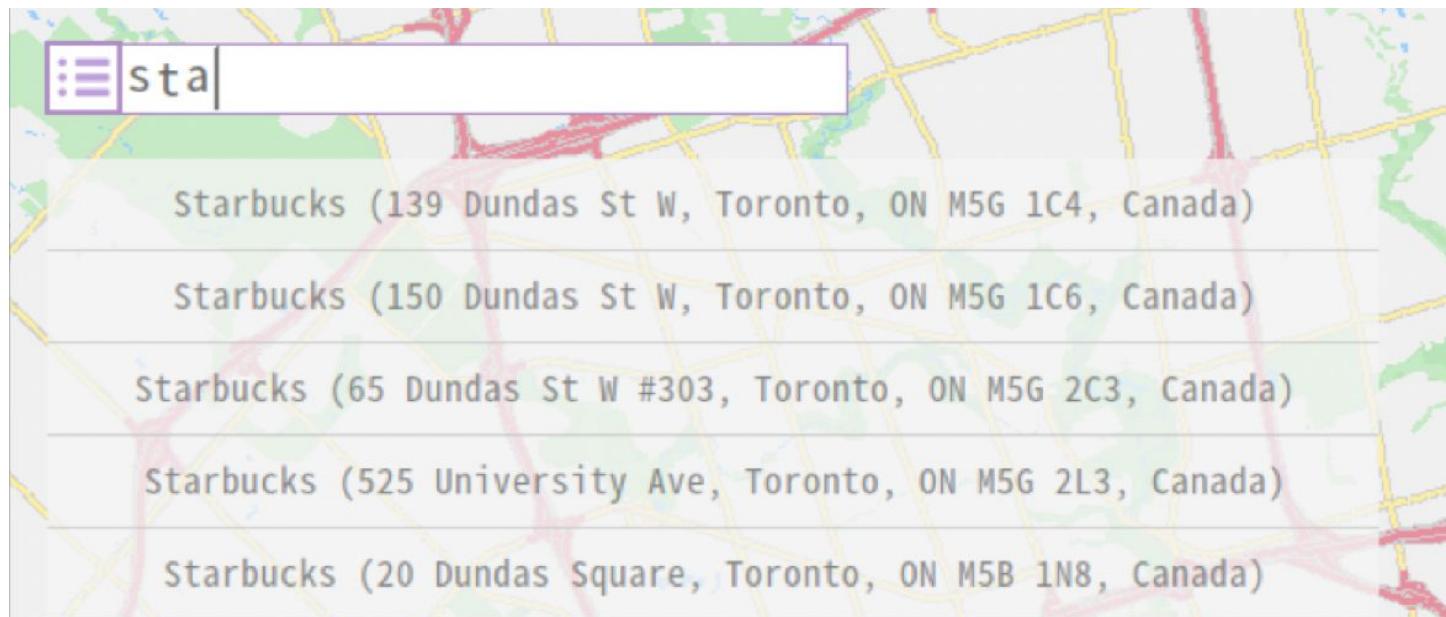
# POI

- Menu



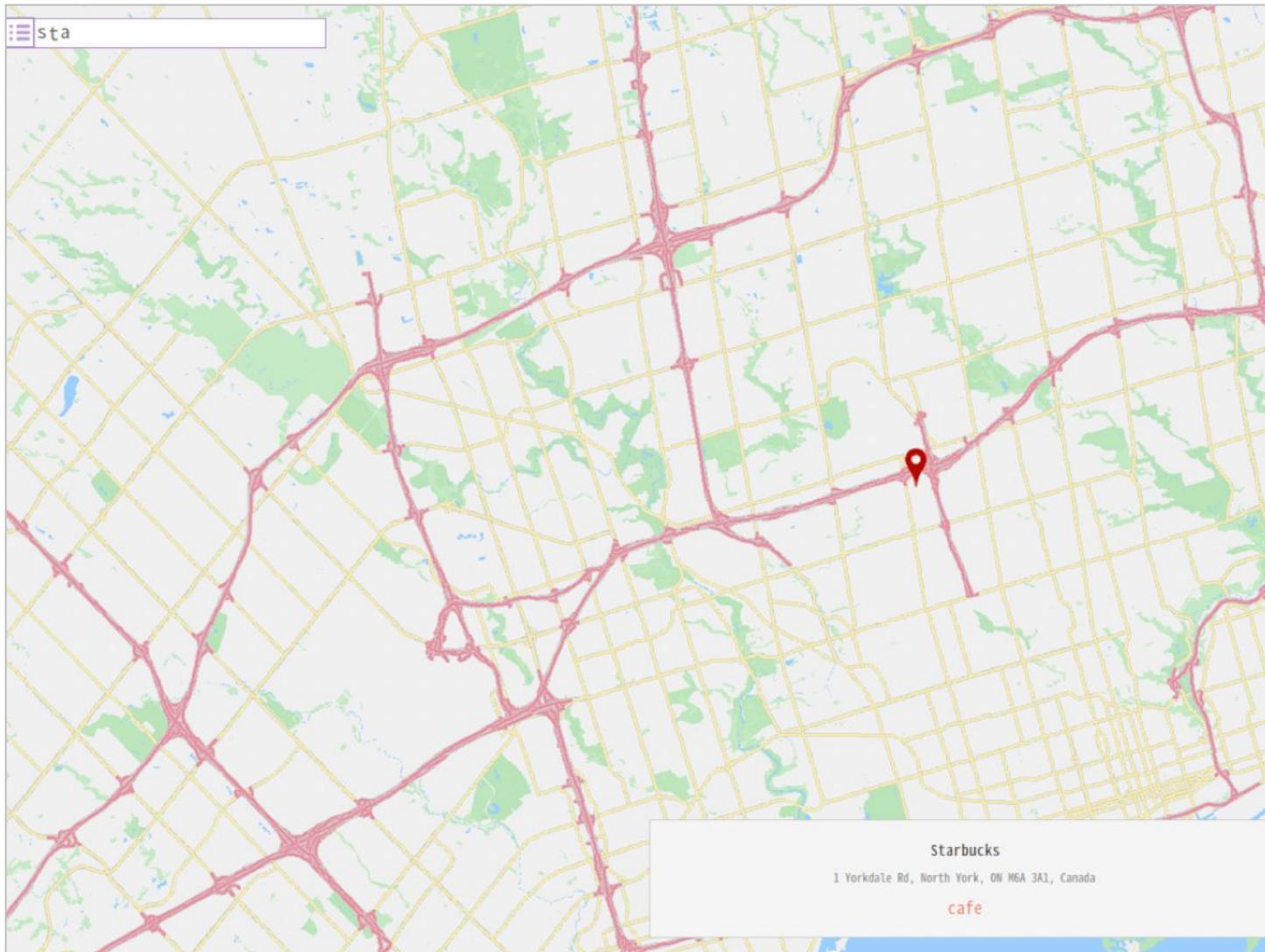
# POI

- Search bar



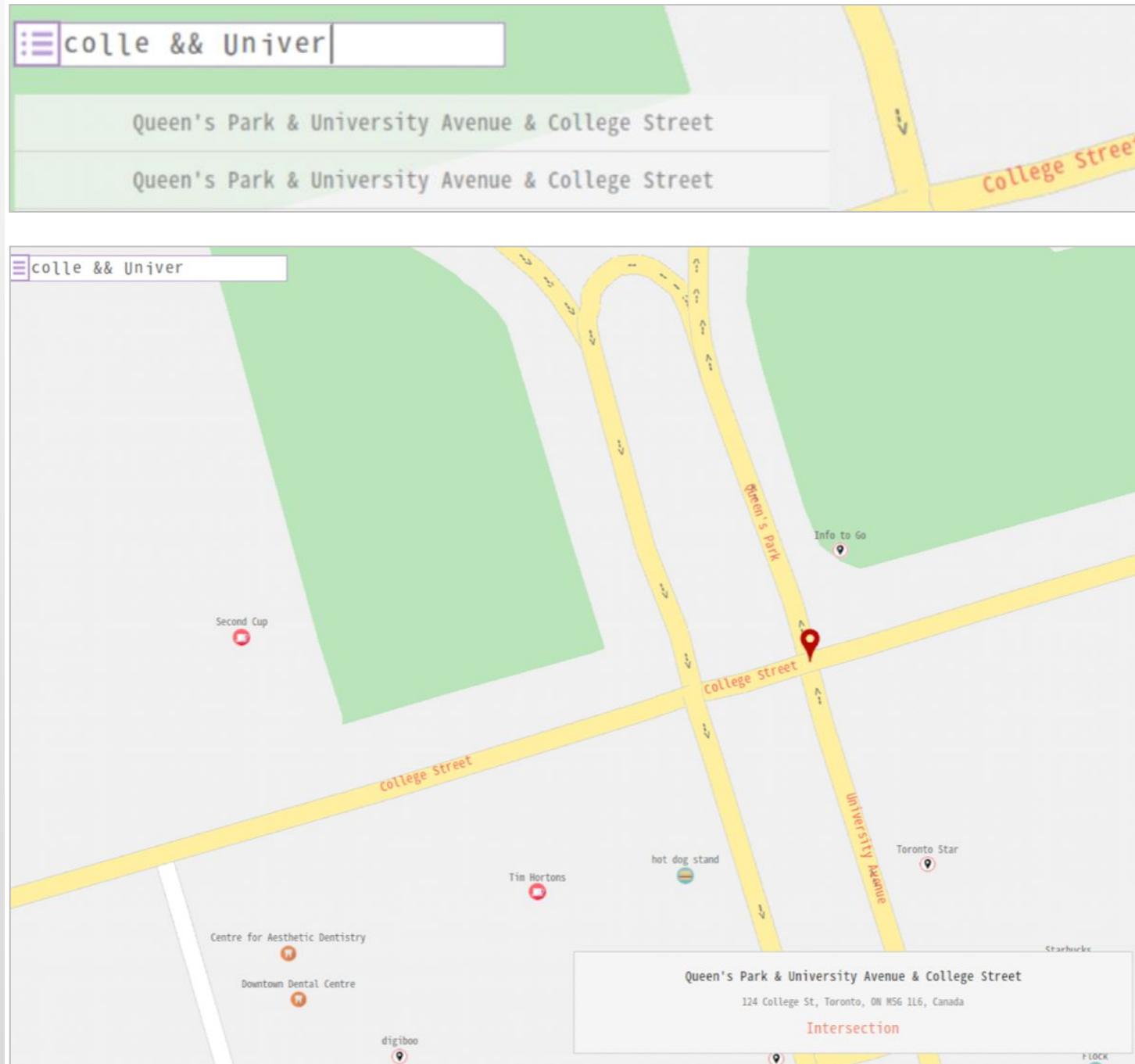
# POI

- Search bar



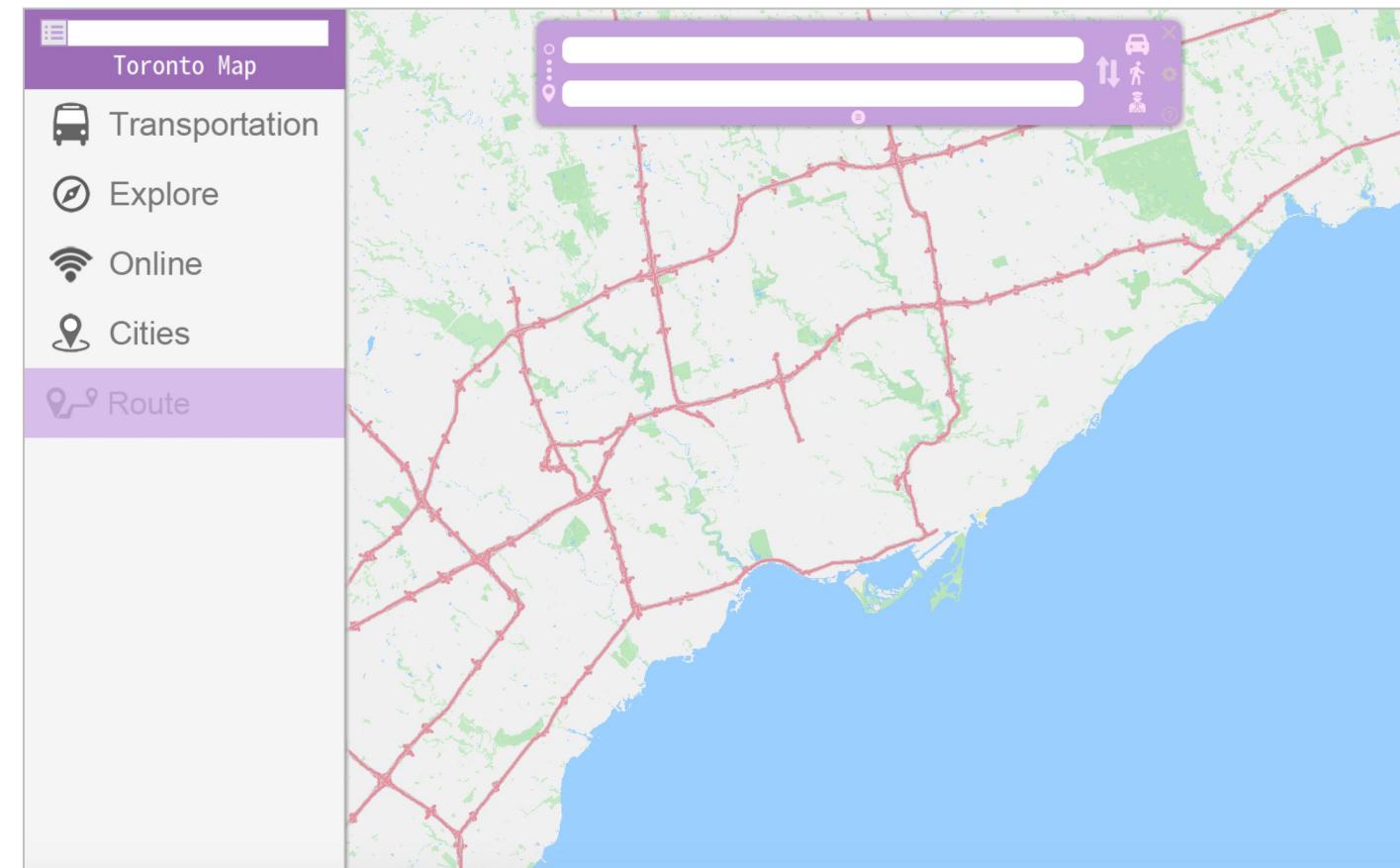
# SEARCH BAR

- street name && street name
- returns an intersection

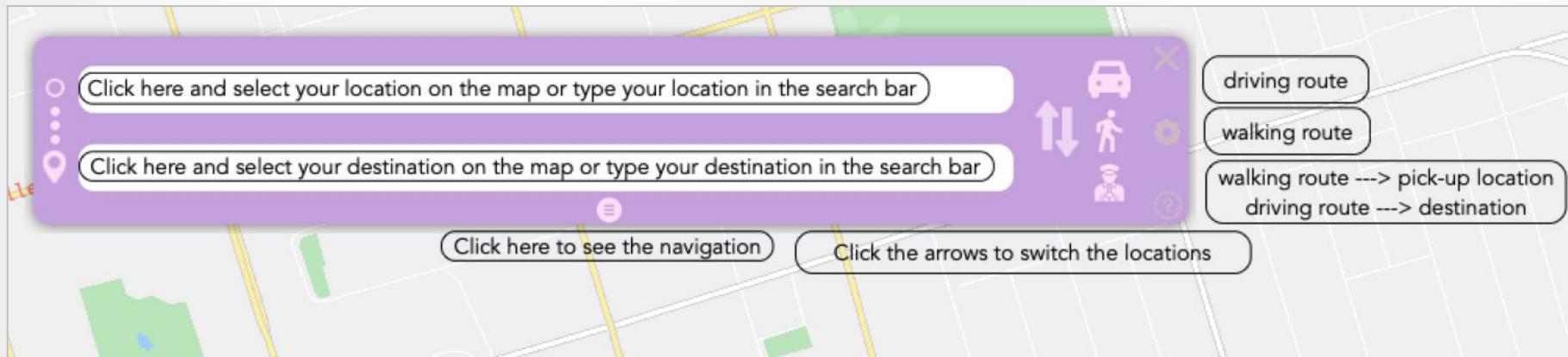


# ROUTE SEARCH DESIGN

Menu → “Route”



-  help and instruction

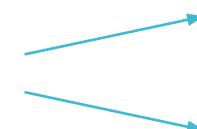


# HELP AND INSTRUCTION

# ROUTE SEARCH BOX



2 search bars



Start Location

Destination

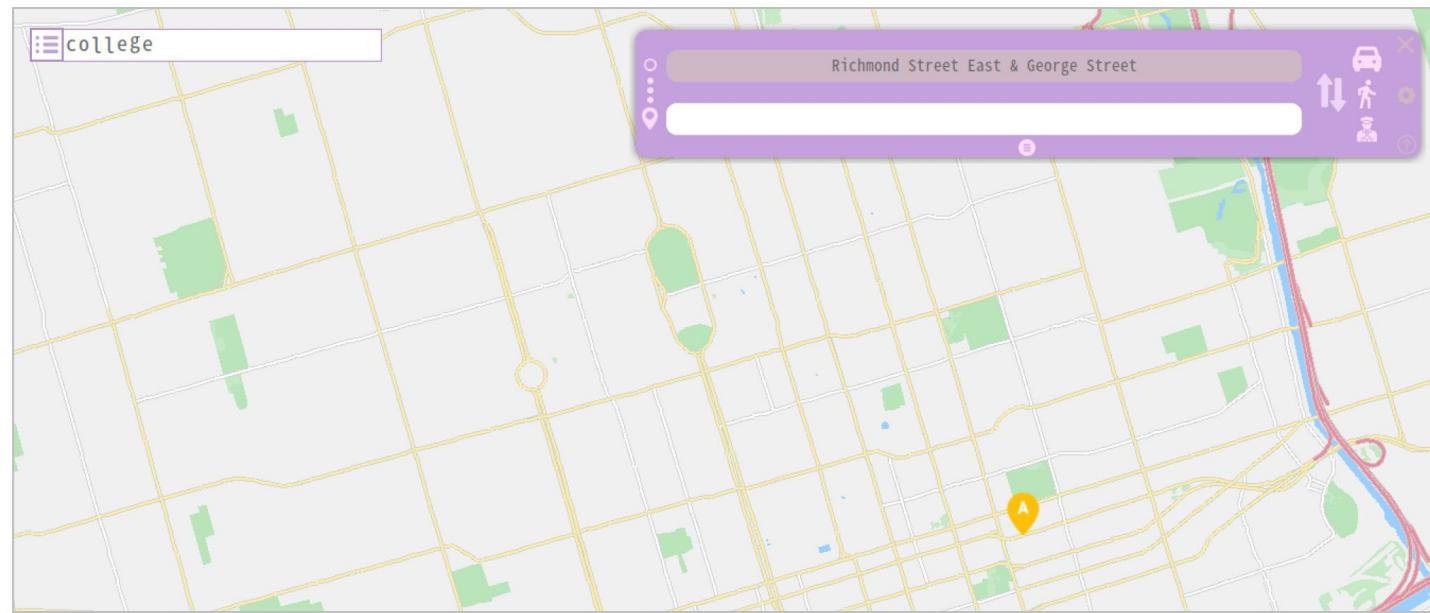
# START LOCATION

- Click the start location search bar.
- Type the street name in the main search bar.

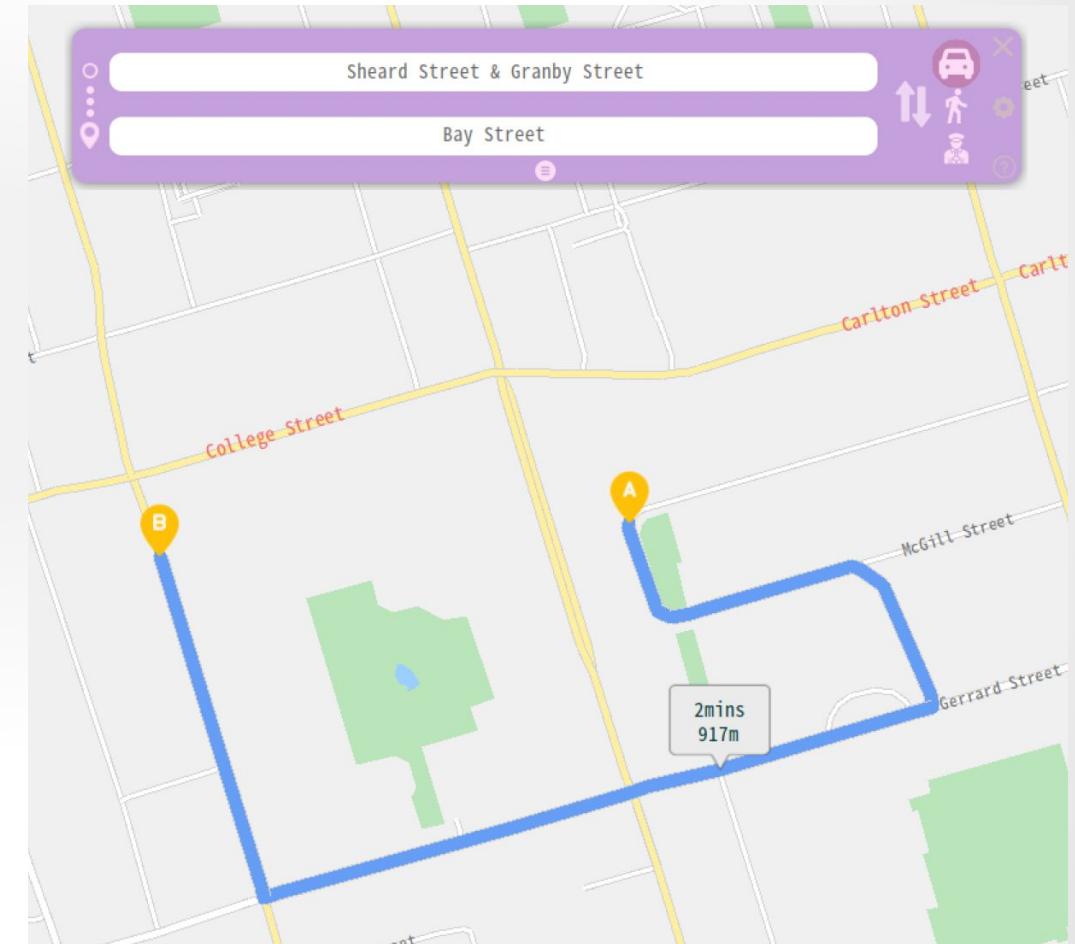
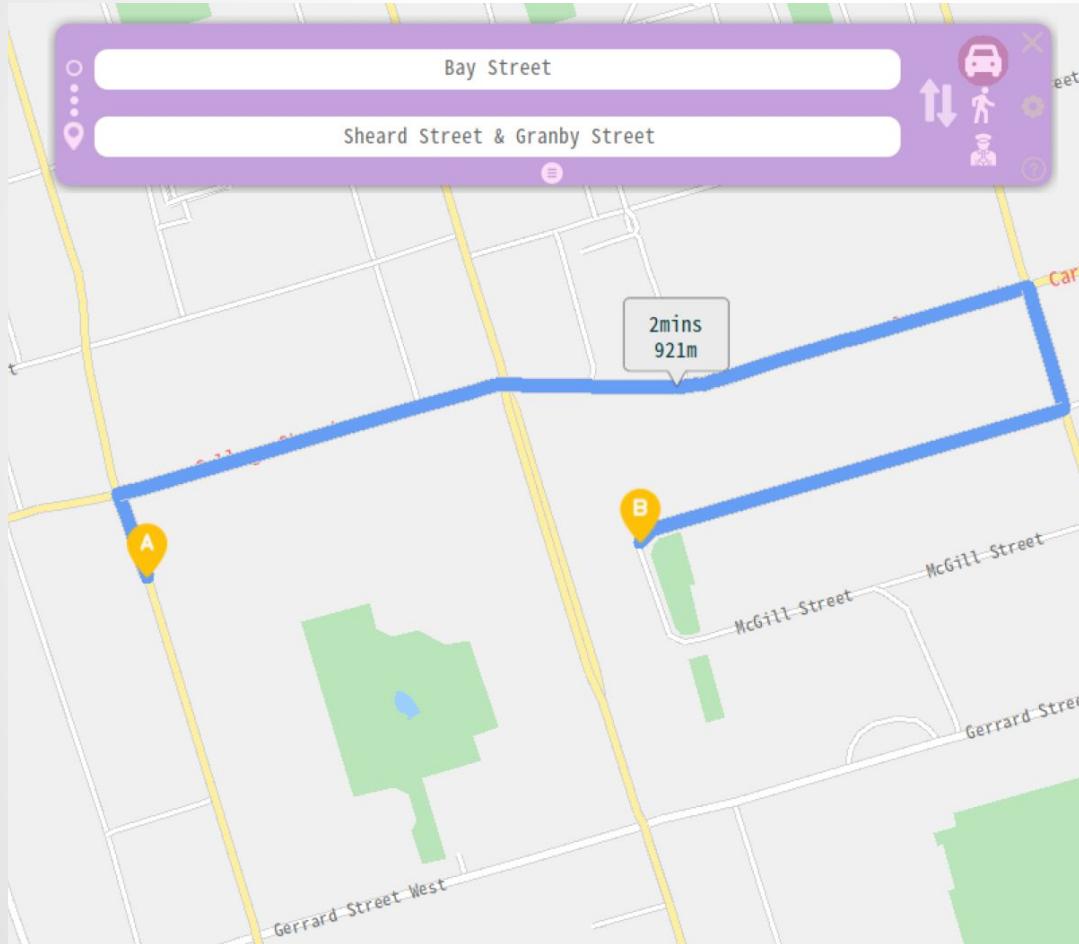


# START LOCATION

- Select the location.
- Press “Enter”.

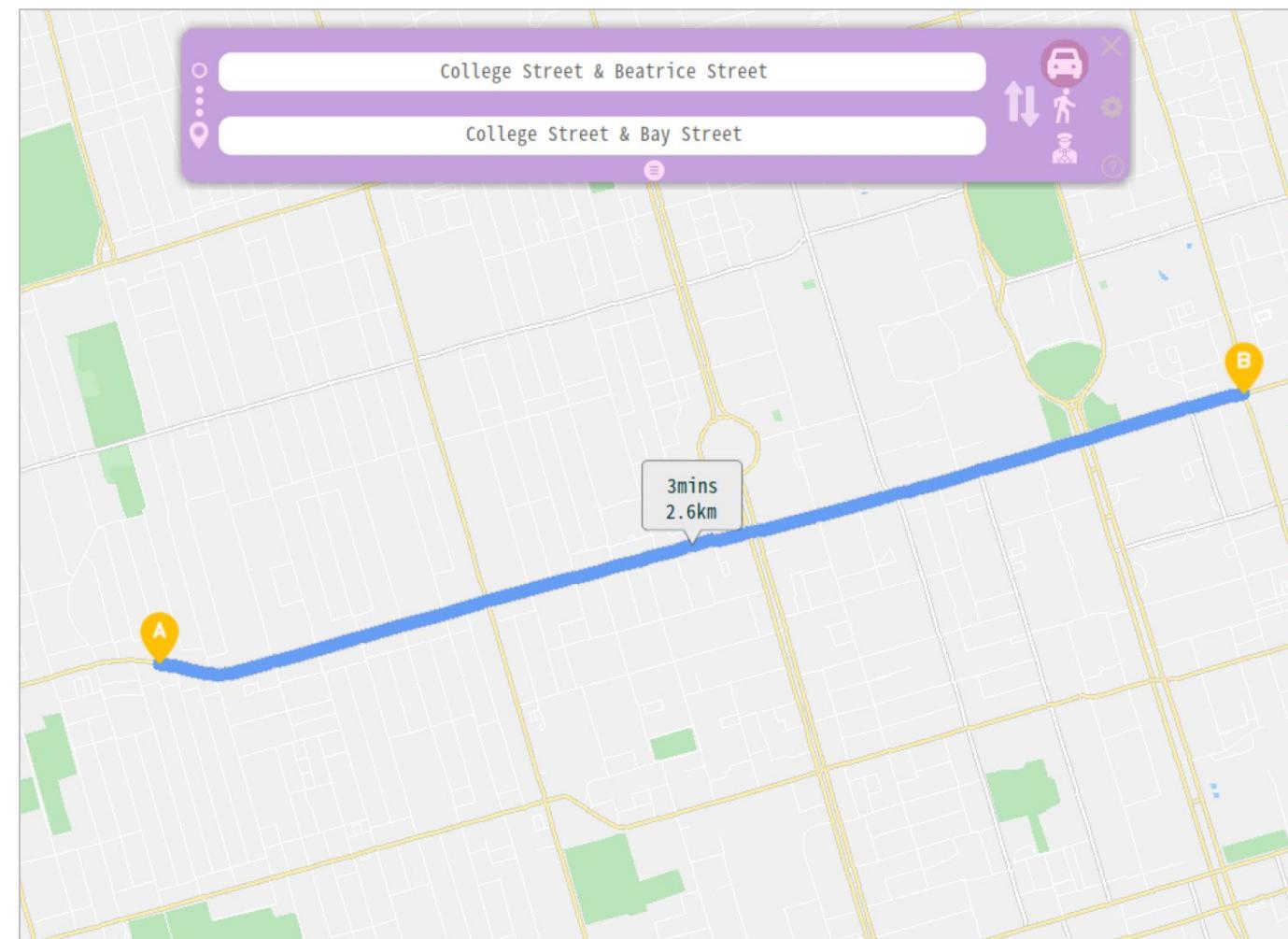


# SWITCH THE LOCATIONS



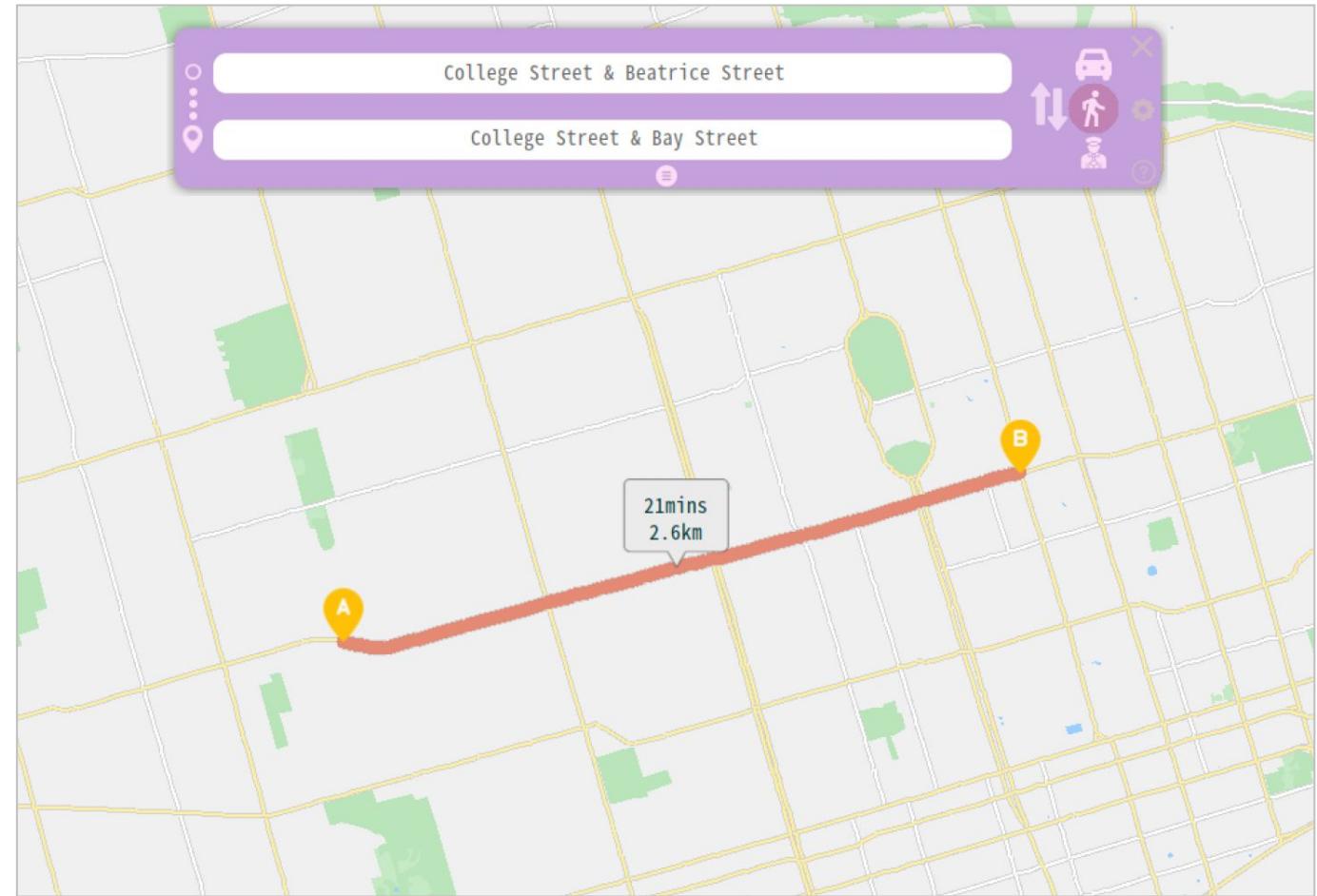
# DRIVING PATH

-  “car” logo



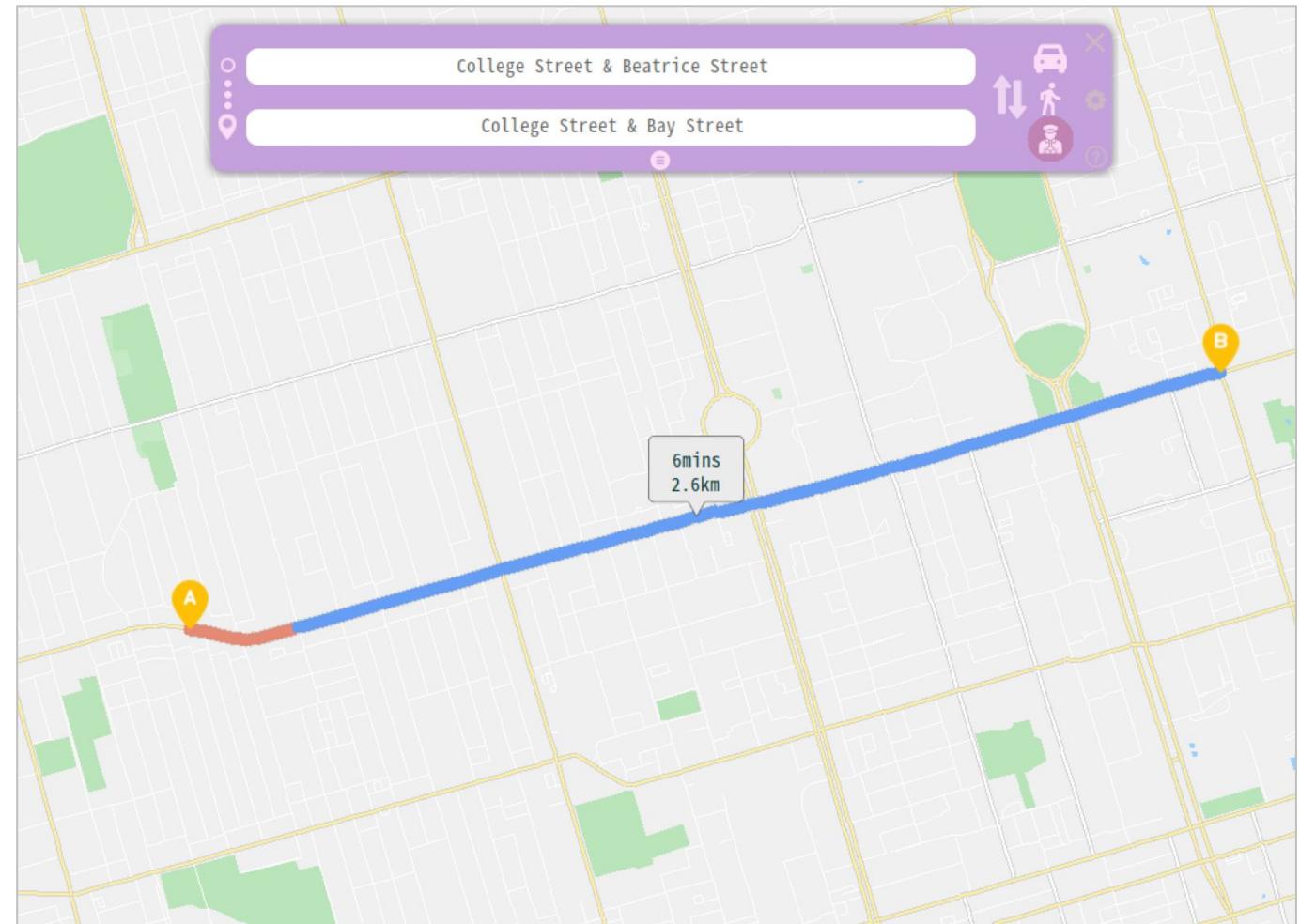
# WALKING PATH

-  “pedestrian” logo



# PICK-UP

- walking + driving
-  “driver” logo

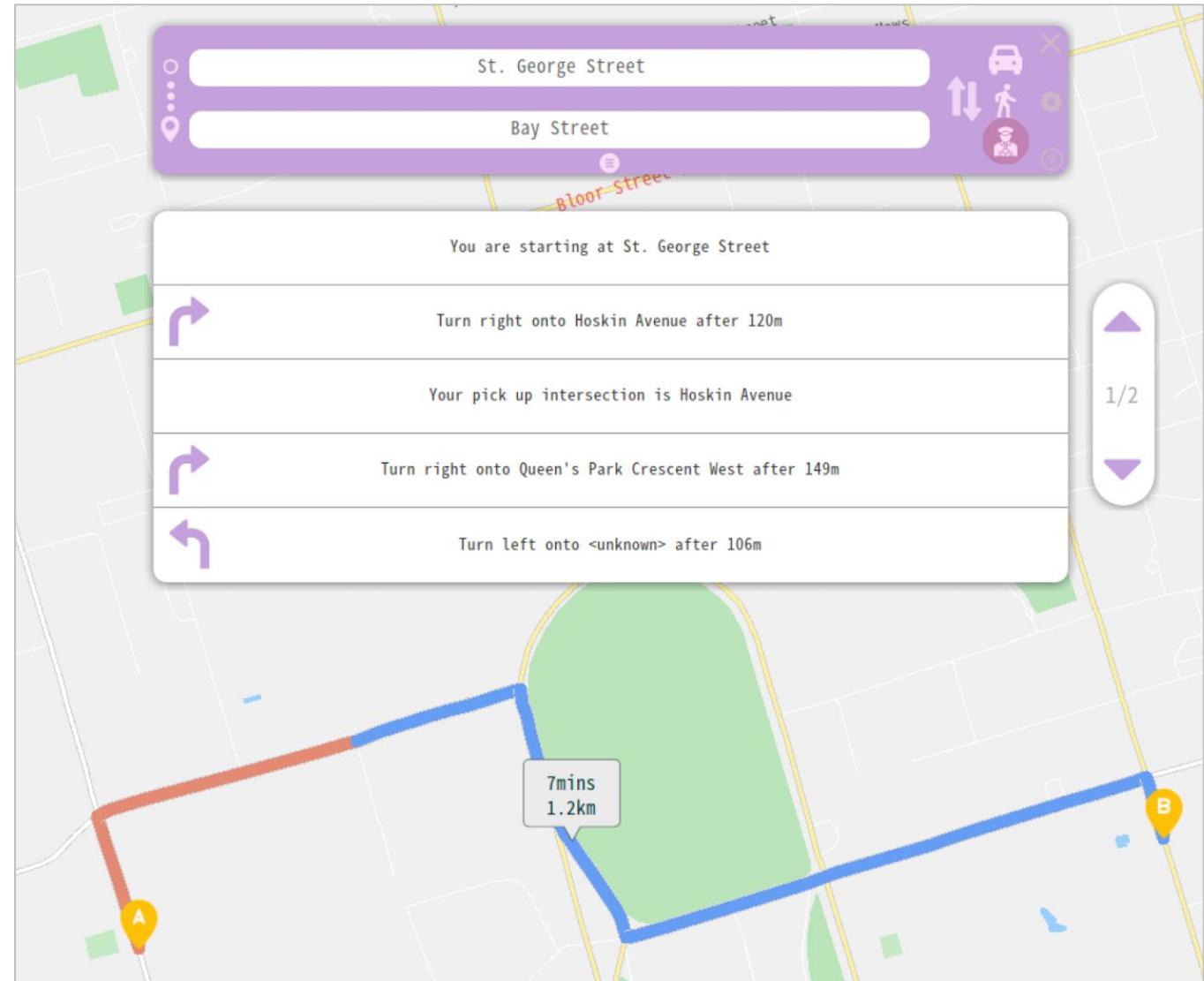


# SETTINGS



# NAVIGATION

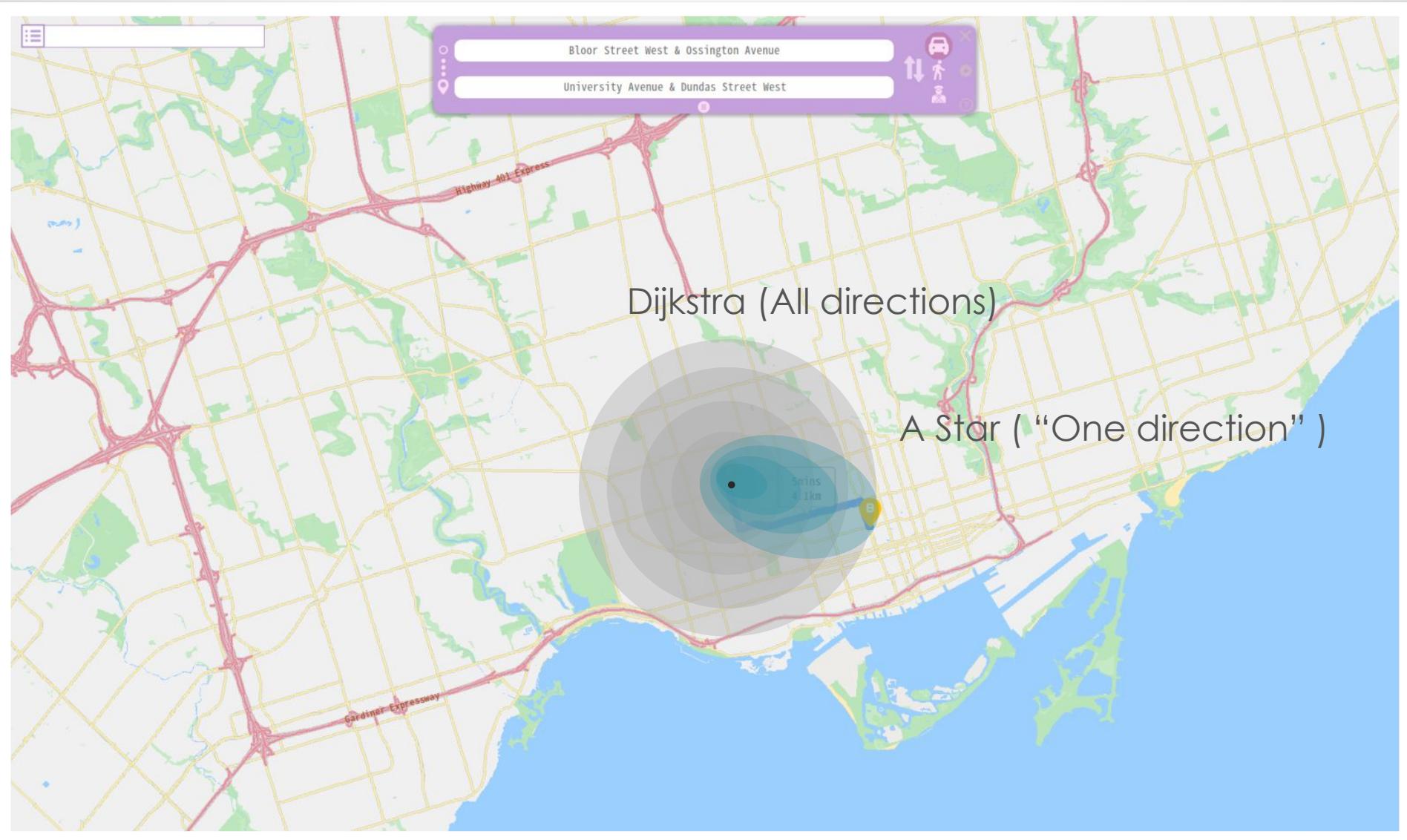
- Click the button-middle button.
- drop - down list
- intersection information
- page turning



# Path Searching Algorithms and Optimizations

- Dijkstra vs A star
- Multi-Source vs Single-Source

# Path Searching Algorithm



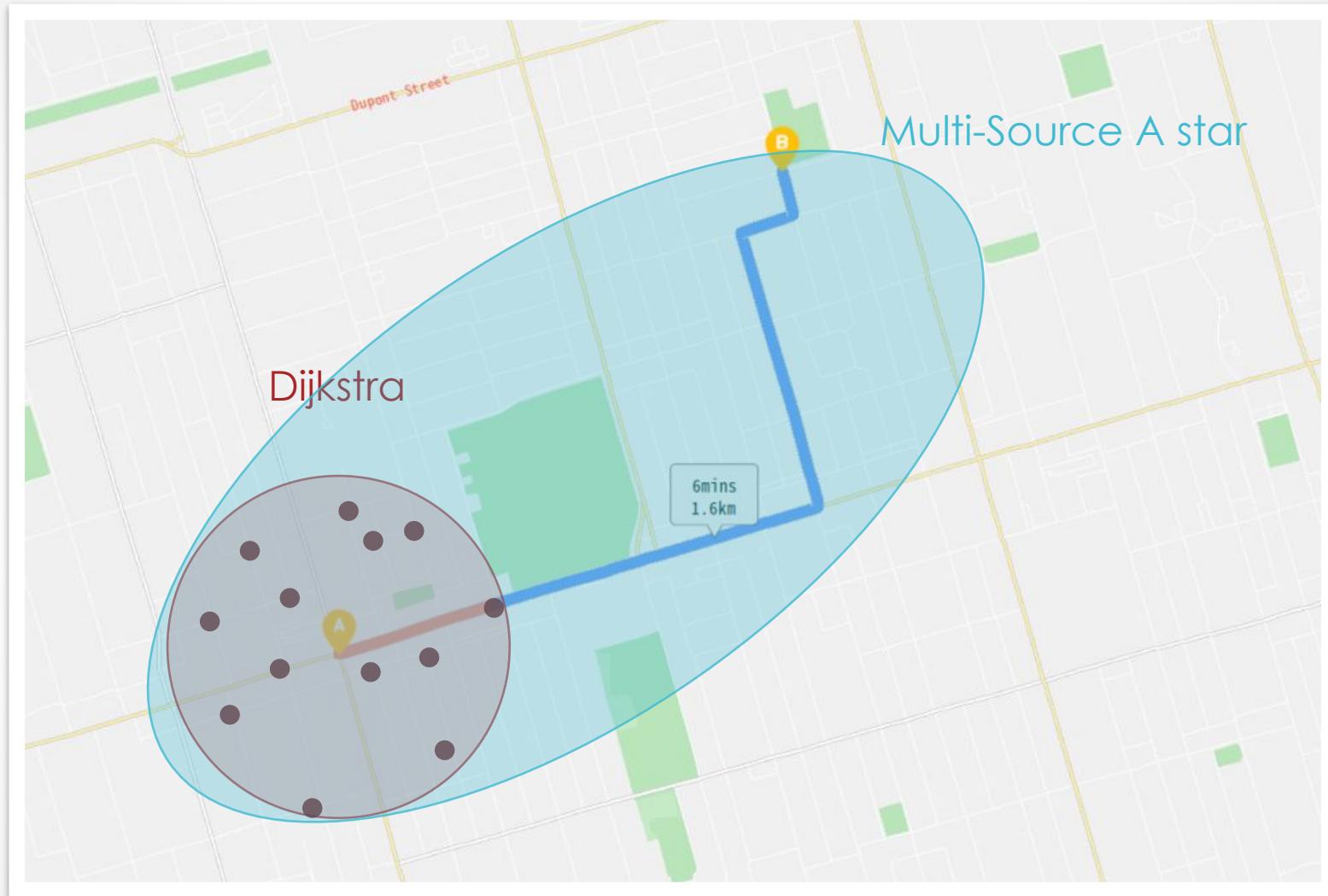
# Dijkstra and A star

## Performance Comparison

City\Algorithm	Dijkstra	A Star
Toronto	~ 0.1 s	~ 0.06s
New York City	~ 0.12s	~ 0.06s
London	~ 0.13 s	~0.07s

Path searching time comparison between Dijkstra and A Star in three different cities

# Walk to Pickup Algorithm



# Multi-Source and Single Source

## Performance Comparison

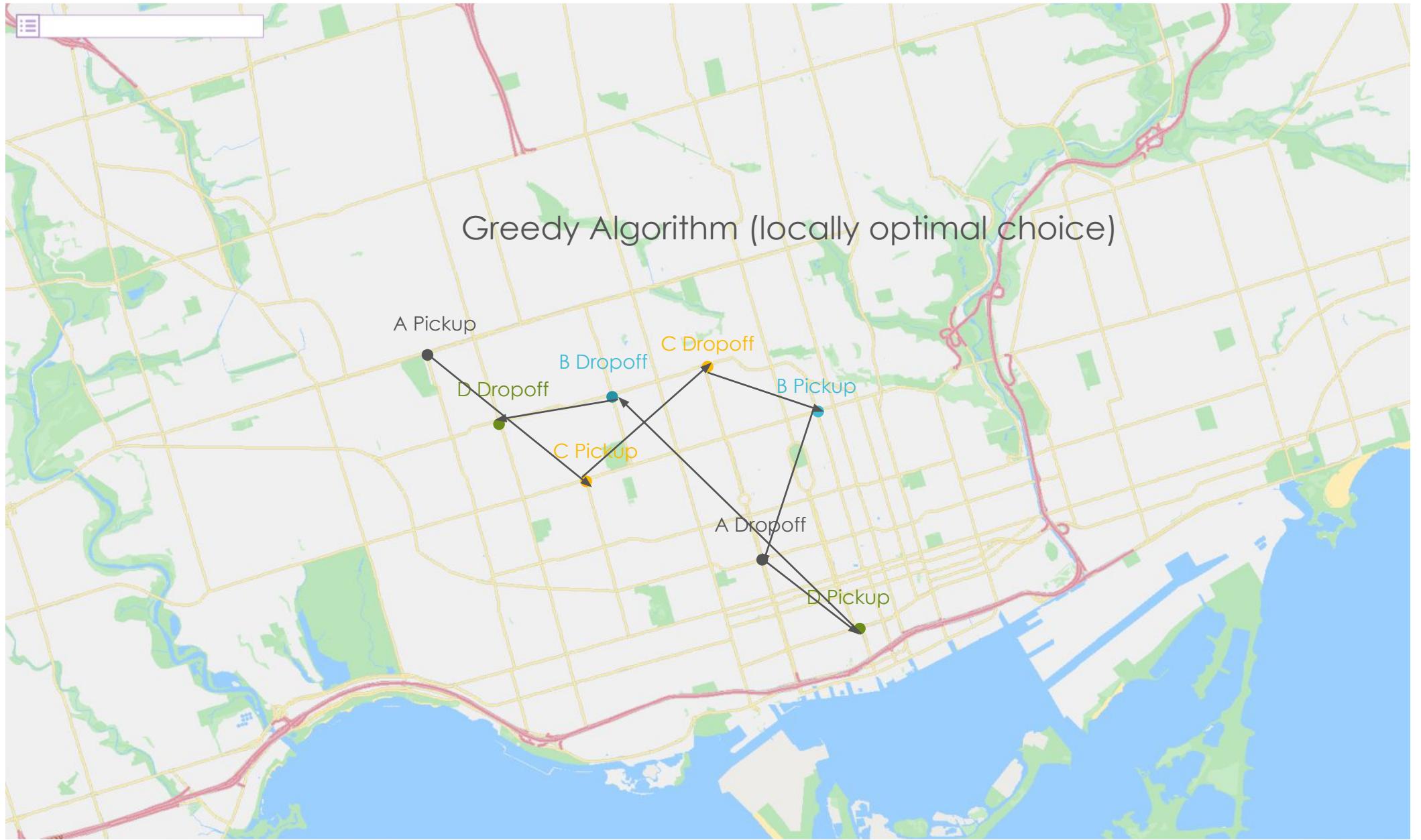
City\Algorithm	Multi-Source	Single-Source
Toronto	~ 0.1 s	~ 0.78s
New York City	~ 0.11s	~ 0.94s
London	~ 0.09 s	~0.82s

Path searching time comparison between Multi-Source A star and Single-Source A Star in three different cities

# Courier Path Optimizer

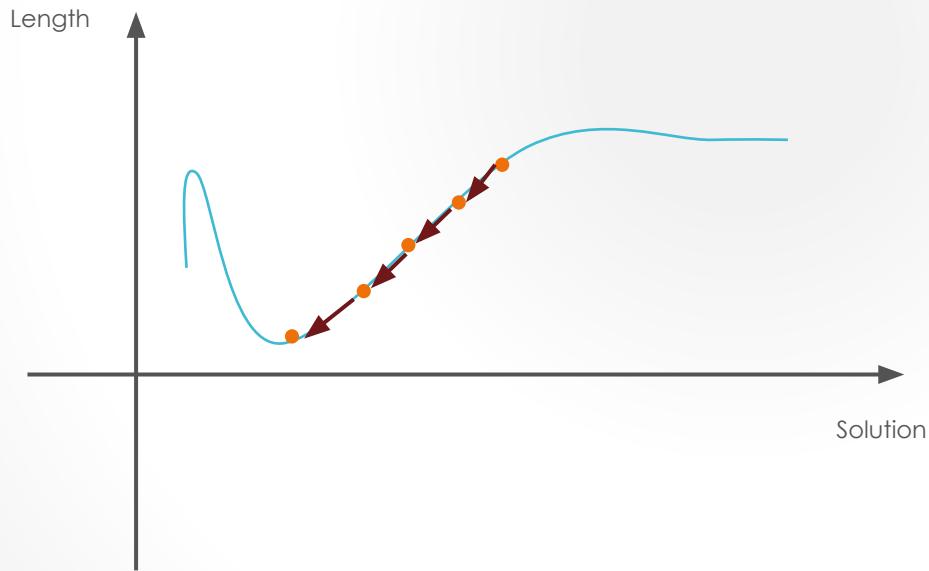
Variant of TSP(Travelling Salesman Problem)

- Greedy Algorithm
- Iterative Algorithm
- Simulated Annealing



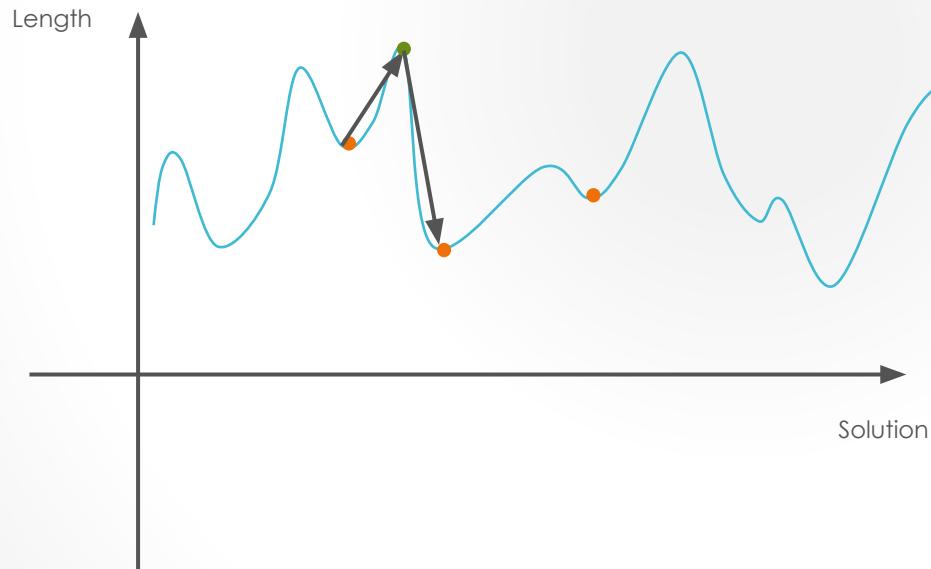
# Improving Greedy Algorithm with Iterative Optimization

- We need a perturbation function to change the current result
- Accept new result only when it is better than previous version
- Keep changing the result for many times

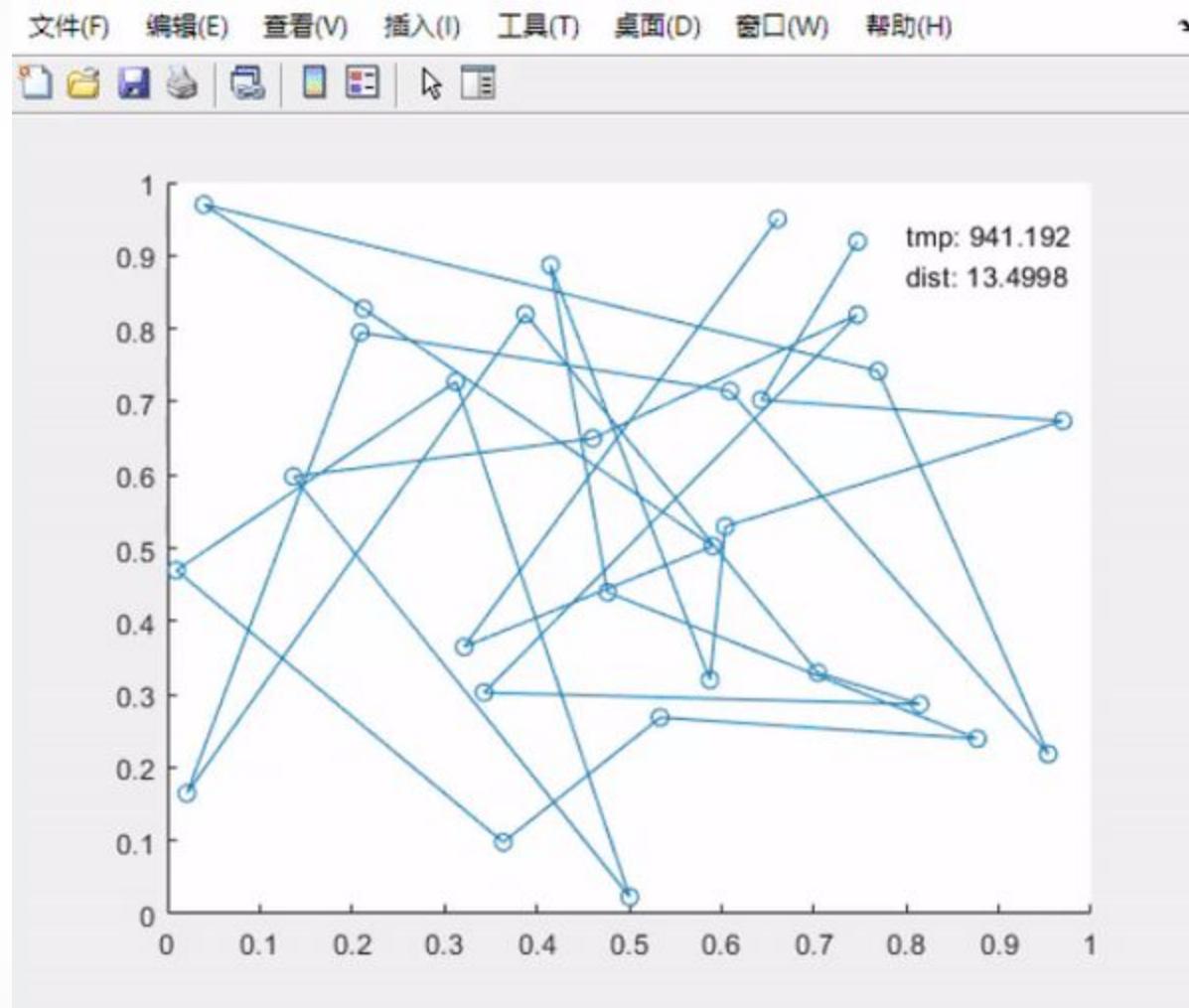


# Using Simulated Annealing to jump out of local minimum

- In practice, there are not only one minimum but multiple
- Accepting only good result will get stuck in those local minima
- Need to accept bad results with some probability

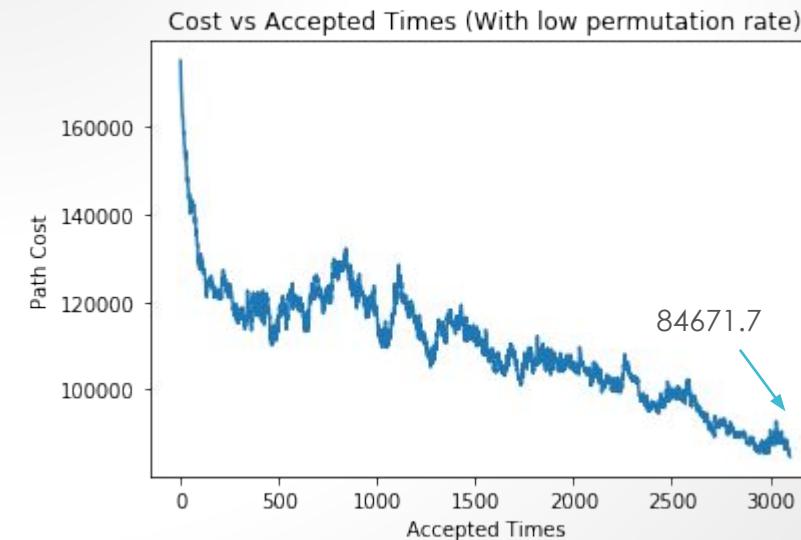
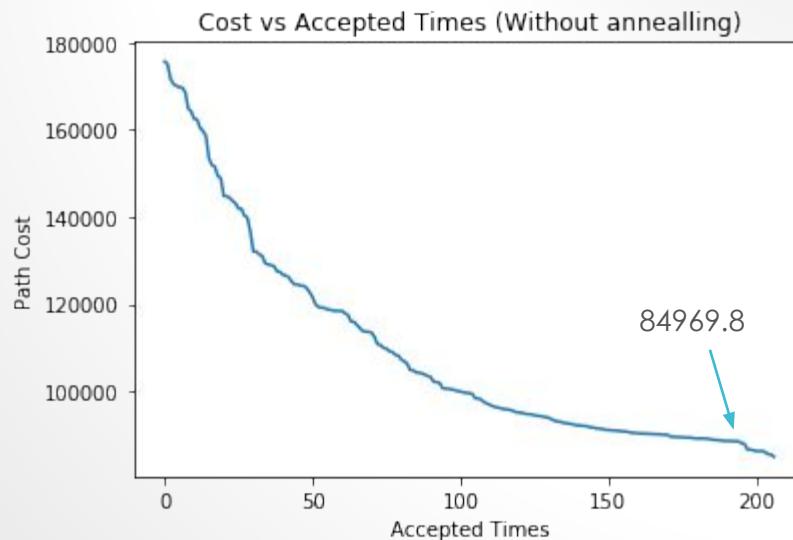
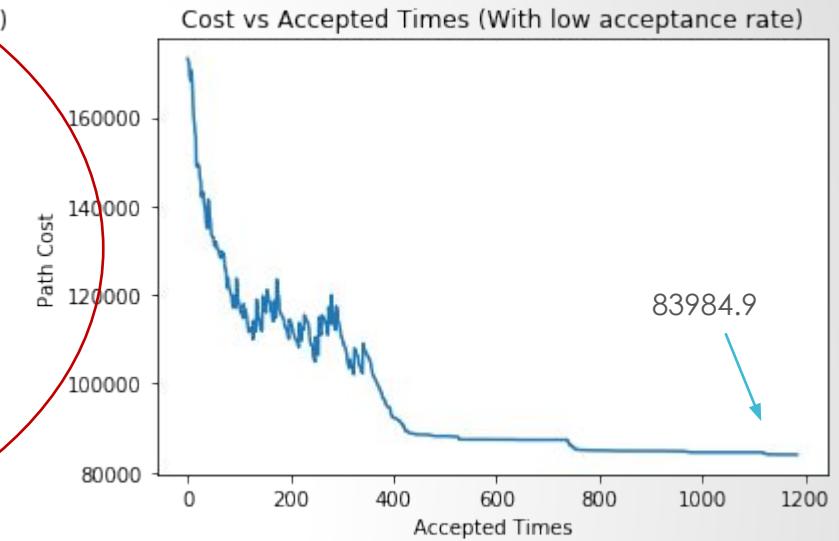
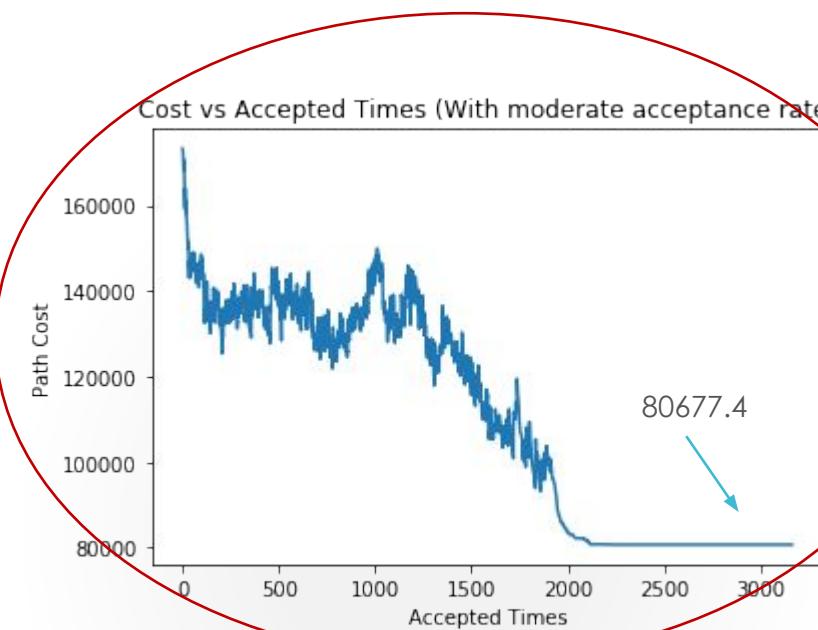
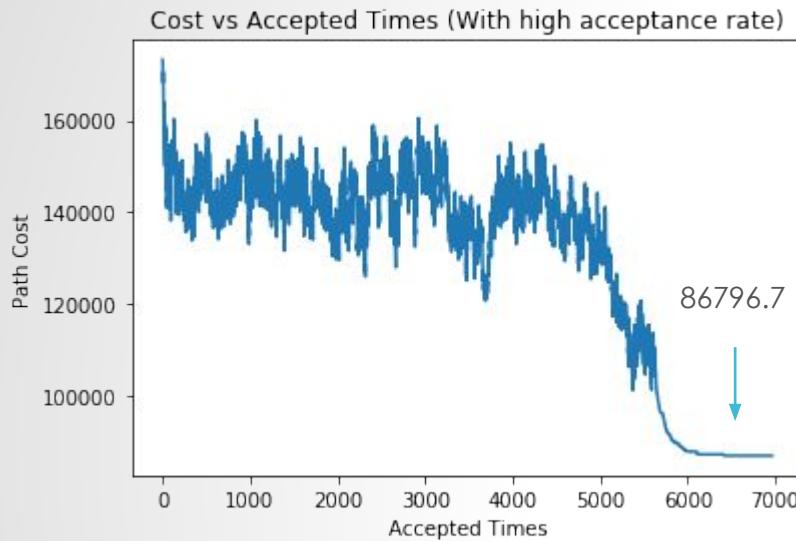


## Demonstration of Simulated Annealing using MATLAB



# Tuning the hyperparameters

- Permutation Rate
- Acceptance Rate



## Story about our perturbation function

How to get different permutation rate ?

Perturb( Perturb( Perturb(.....) ) )

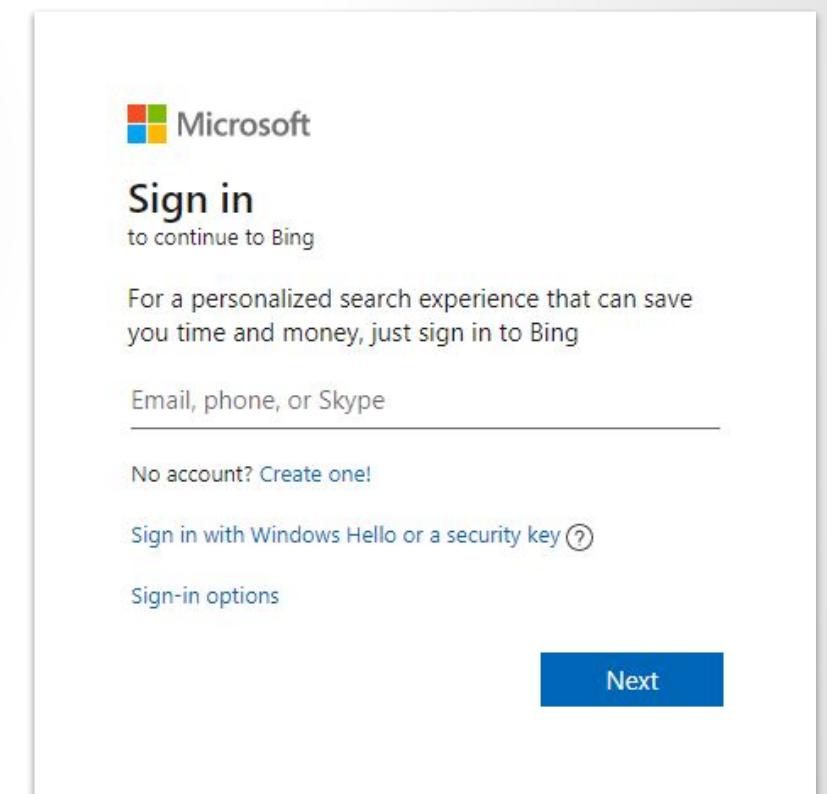
Lack of Memorability [1]

For the future .....

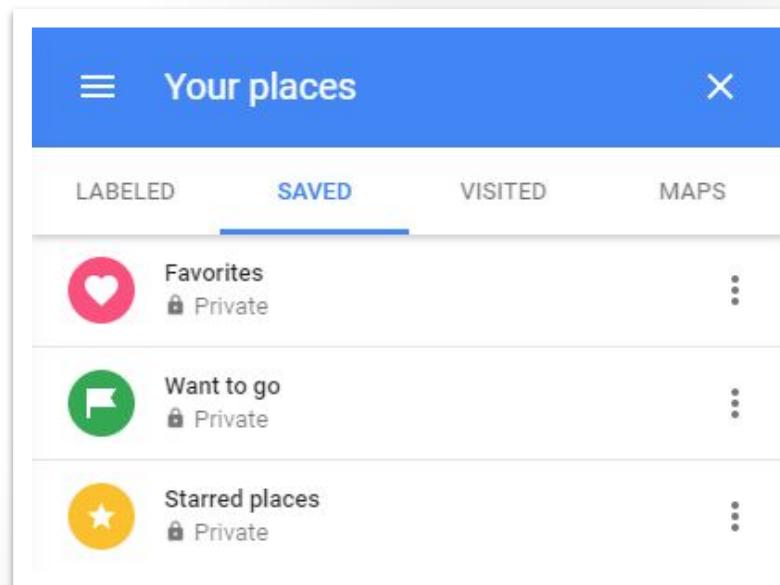
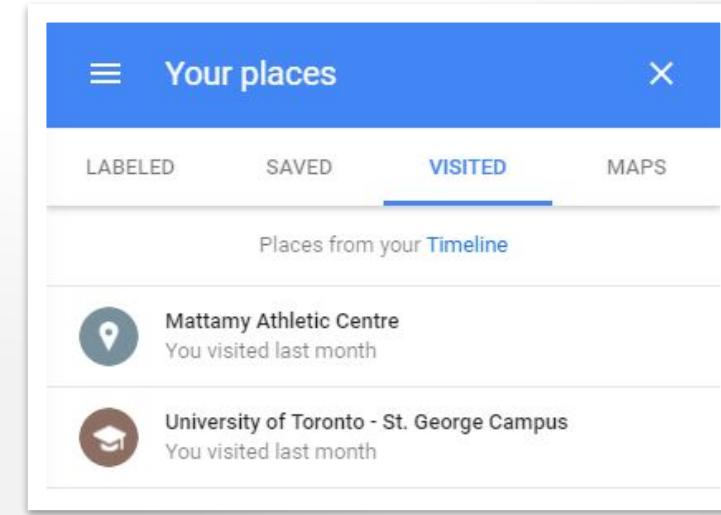
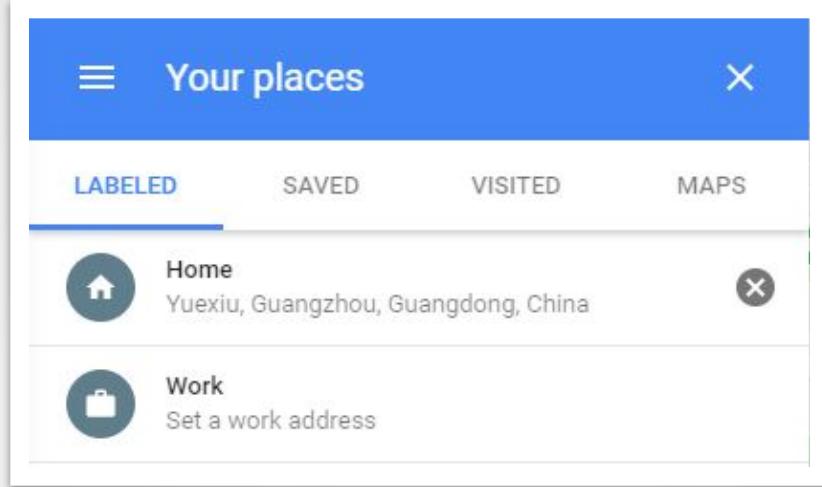
## The account system

- Users can create their account and login
- Users can mark their interested places
- Users can personalize their account (Language, theme)

Login UI Example from Bing Map



## Current Solution on the market (Google map)



We have previous experience on network connection

## **How to build the account system ?**

Transmit data to the server and fetch from it

Rent an online server (VPS)

Create account database on the server

Data encryption

We can do more .....

- Use machine learning to group different users
- Provide different groups with different information based on their interest
- Develop Android/IOS version of our map



Take away

