Toy Maps

This document contains information on 3 toy maps for debugging (queens_park.osm, robarts.osm, baldwin.osm). You should use this document to check whether your algorithm is producing results as you are expecting. Note: these maps are much simpler than toronto.osm and toronto_full.osm, therefore, getting the correct results for these maps does not guarantee that your algorithm is correct.

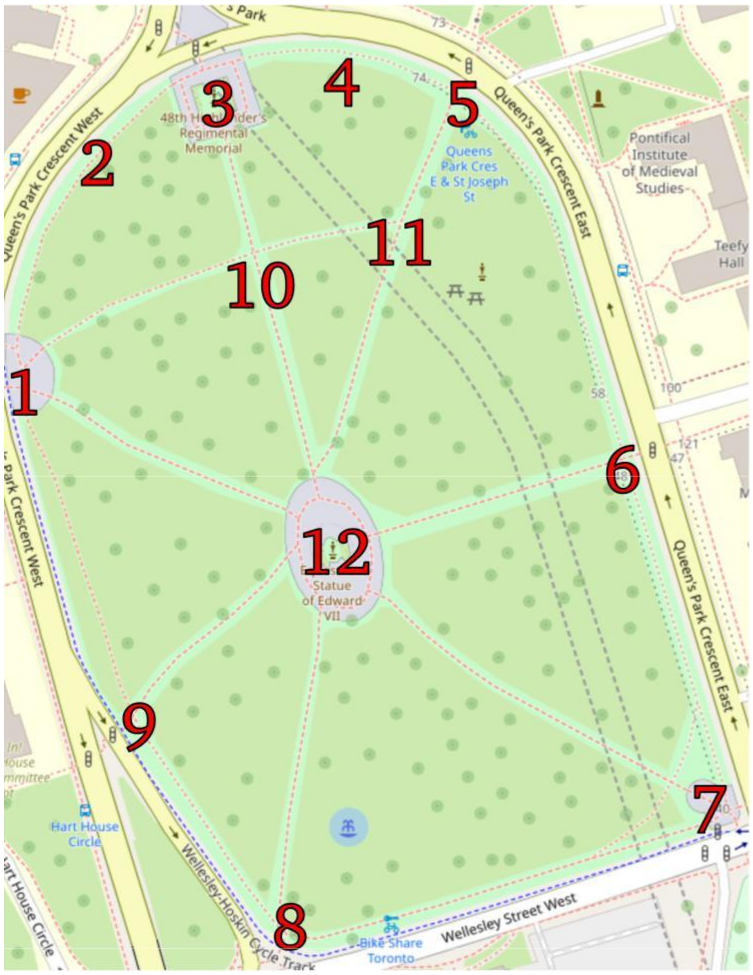
The following information are included for each toy map:

1. A visualization of the map with labeled nodes
2. A table containing the optimal paths for some examples of sourceNode and destinationNode when **CostFunctionAllFeatures** is used. Using other cost functions may result in different optimal routes.

How to use the toy maps for debugging:

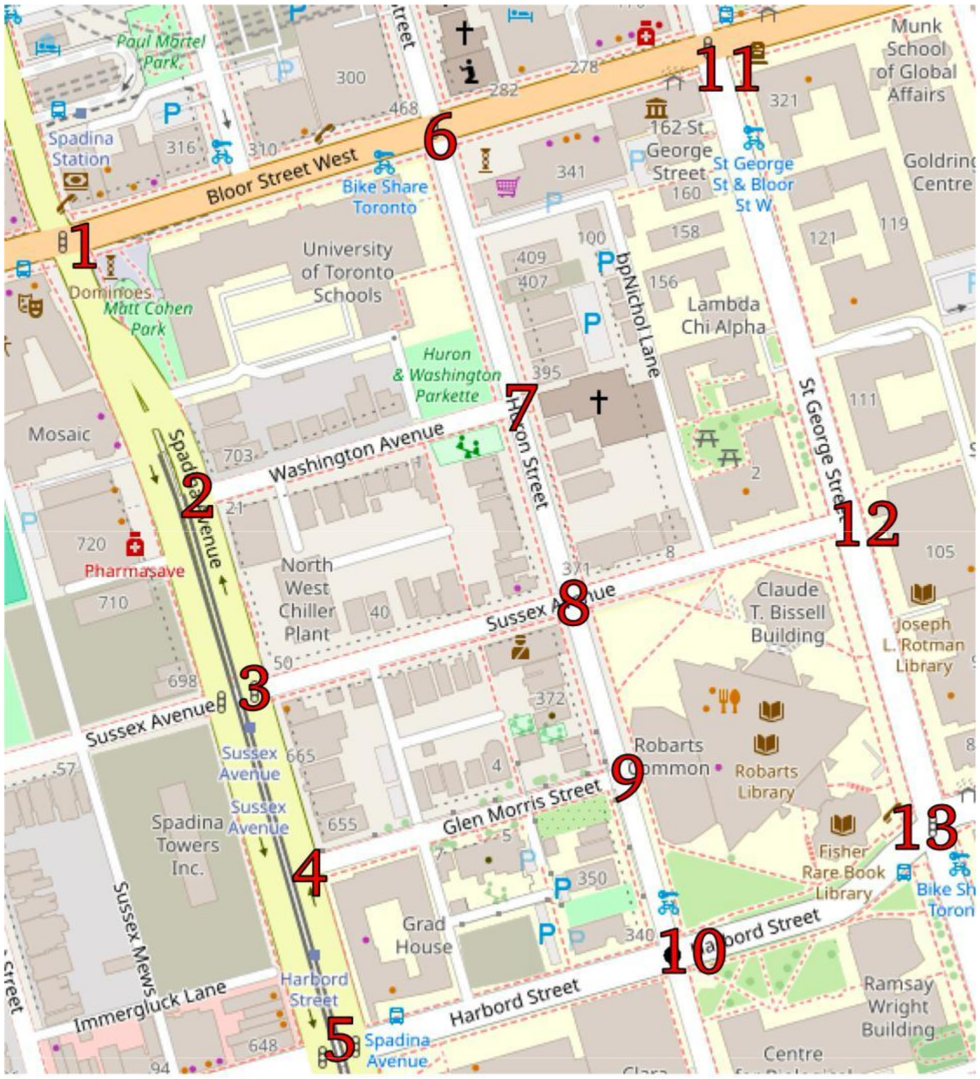
1. Make sure CostFunctionAllFeatures is used in Demo.java to calculate the cost of a path.
2. Load a toy map by editing the variable *osmFile* in main function of Demo.java
3. Set the start and destination waypoints as seen in the table entries and observe if your solution matches the optimal solutions given. (note that not all algorithms are guaranteed to find the optimal solution)

queens_park.osm


1
2
3
4
5
6
7
8
9
10
11
12
Queen's Park Crescent West
Queen's Park Crescent East
48th Highlander's Regimental Memorial
Queens Park Cres E & St Joseph St
Pontifical Institute of Medieval Studies
Teefy Hall
Statue of Edward VII
Hart House Circle
Wellesley-Hoskin Cycle Track
Wellesley Street West
Bike Share Toronto

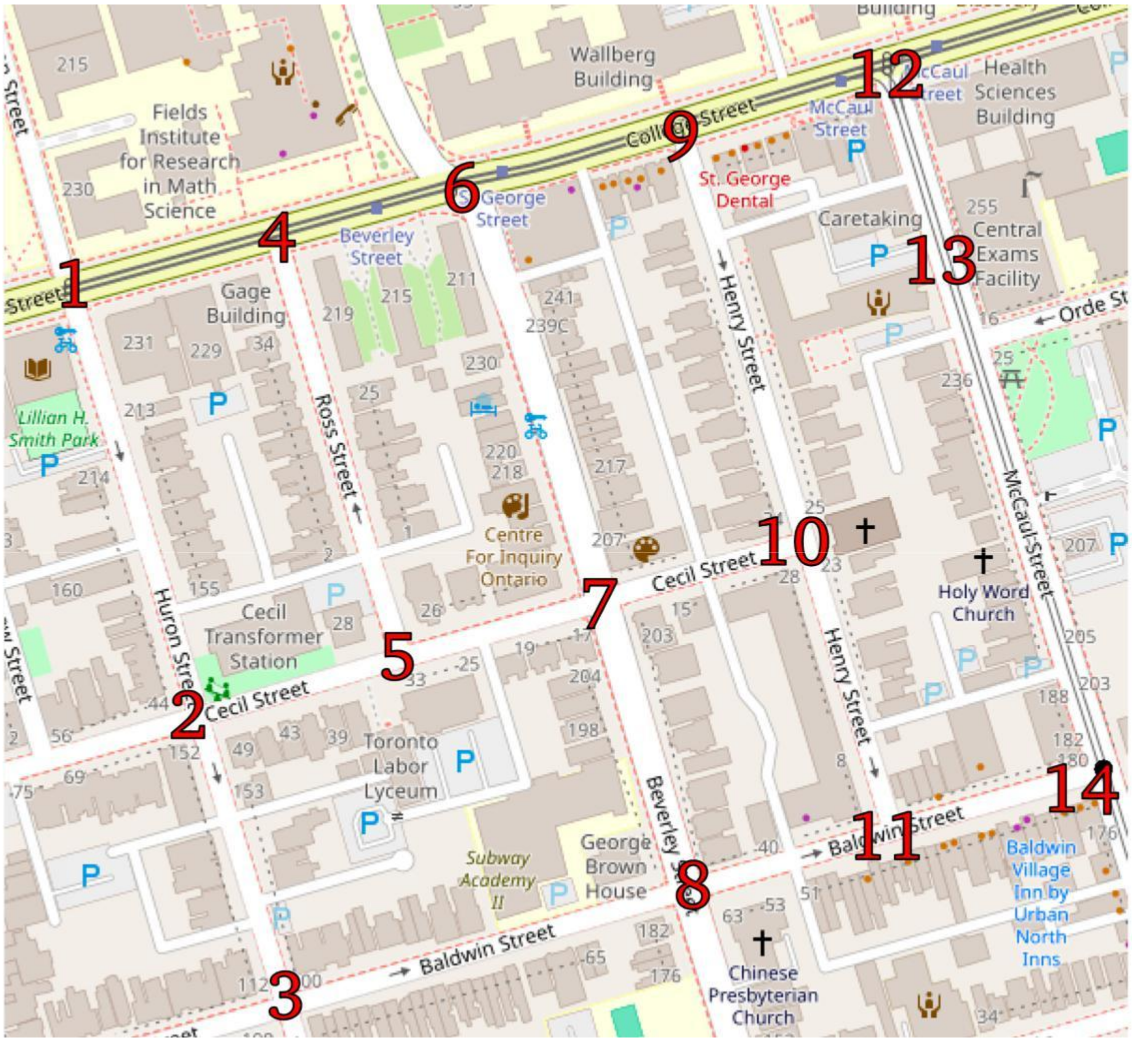
Source node	Destination node	Optimal Path
1	7	1->12->7
2	11	2->3->10->11
8	4	8->12->10->3->4
6	2	6->5->4->3->2
7	2	7->12->1->2

robarts.osm



Source node	Destination node	Optimal Path
5	11	5->10->13->12->11
11	4	11->12->8->3->4
1	13	1->2->7->8->12->13
3	6	3->2->7->6
10	12	10->13->12

baldwin.osm



Source node	Destination node	Optimal Path
13	5	13->12->9->10->7->5
9	14	9->12->13->14
1	14	1->4->6->7->10->11->14
3	12	3->8->7->10->9->12
2	6	2->1->4->6