

## 3.15

Andrew Lee

March 14, 2017

The police department in the city of Computopia has made all streets one-way. The mayor contends that there is still a way to drive legally from any intersection in the city to any other intersection, but the opposition is not convinced. A computer program is needed to determine whether the mayor is right. However, the city elections are coming up soon, and there is just enough time to run a linear-time algorithm.

(a) Formulate this problem graph-theoretically, and explain why it can indeed be solved in linear time

Let all the intersections be the vertices of this graph, and let the valid legal paths between two intersections directed edges  
Then we can solve this as a graph problem using a customized search algorithm.  
What we will use is modified depth-first search. With the following steps:

- I) Arbitrarily pick any intersection, and follow one the set of edges that go in a “similar direction” until you can go no further
- II) Then that intersection where one can no longer go any further must be within the perimeter of the city edge circuit. (The city edge circuit is the abstracted term that refers to the all the vertices that do not have 2 inpoints and 2 outpoints)
- III) Following a similar search pattern as (I), just look for vertices that have less than 1 outpoints along the perimeter. As soon as there is an edge with 0 outpoints, that means that that vertex is a sink in the graph. From a sink, you cannot legally go to any other vertex on the graph, and hence, the mayor is lying.

This algorithm will always work for all scenarios with 4 or more vertices because it can take any arbitrary vertex to start because all it needs to do is to follow the path to the city perimeter, and to follow all edges in the same direction until you reach a sink.

This algorithm will run in linear time Let  $n$ =the number of vertices in the city.  
Then the best case scenario for the algorithm to find an edge from an arbitrarily picked edge is  $n$  since if all the vertices are laid out in a one straight line, then it will be an  $n \times 1$  graph. Starting from one end to the other, we will traverse all points before we can traverse no longer, but at this point, we know that the vertex will be a sink. For the average case that the city is roughly symmetrical, then the time would take about  $\sqrt[3]{n}$ . The time it would take to traverse all edges would take roughly speaking  $\sqrt[3]{n}$  steps assuming the area is about  $\sqrt[3]{n} * \sqrt[3]{n}$ . Hence, in theory, this algorithm on average will take about  $4 * \sqrt[3]{n} + \sqrt[3]{n}$  steps which is a root function, which is  $= O(n)$ .

(b) Then, the only difference with the above algorithm is that instead of picking an arbitrary vertex, we pick the town hall. The problem can be solved in at most linear time for the same reason as the above.