CS2040S: Data Structures and Algorithms

Recitation 7

Goals:

- Model problems using graphs
- Transform graphs to utilize existing algorithms as black boxes
- Identify and formulate shortest path problems
- Appreciate the representational power of graphs

Problem 1. Empire *State* of Delivery

In the USA and most parts of the world, a right-hand traffic system is enforced (Singapore follows the left-hand traffic system). In the US, when the green light is active for drivers at an intersection, they must either proceed straight or make a left turn. If the red light is active, drivers must first come to a complete stop and should they wish to make a right turn, they may after giving way to oncoming vehicles and when it is safe to do so. This is known as the "right turn on red" rule which generally applies to most intersections across the US unless indicated otherwise (slip roads are not as prevalent in the US as they are in Singapore). Therefore, it is generally the case that vehicles approaching a four-way intersection in the US may turn in all three directions (left, straight, right). However, to address the Vehicle Routing Problem, delivery companies such as the United Parcel Service (UPS) found that minimizing left-turns in their vehicle routes saves them as much as 10m gallons of fuel (20,000 worth of tonnes carbon dioxide emissions) and improved delivery throughput by 350,000 each year, despite the longer routes [1]. The intention of such a strategy is simple: to cut down time spent at intersections waiting for the green light (which also wastes fuel) and to minimize risks of accidents due to turning.

It's summer break, and you are interning at *Manhattan Eats*, a hot food delivery startup operating in New York City. One notable characteristic, amongst many others, of Manhattan is how its roads are laid out to form a uniform grid where roads running North-South are known as *avenues* and roads running East-West are known as streets. An example of Manhattan's intersection-filled layout is shown in Figure 1. Indeed, such a city layout is what makes vehicle routing all the more critical for the company's success — one wrong turn could make a very dissatisfied customer!

[1] The Conversation: "Why UPS drivers don't turn left and you probably shouldn't either"

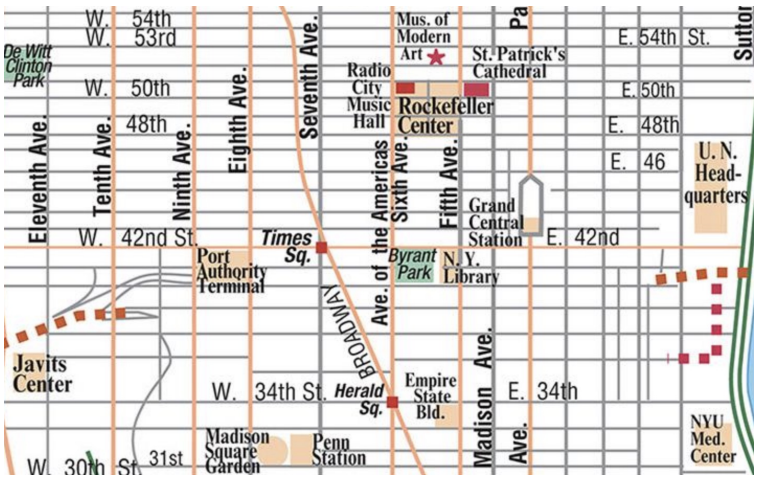

Figure 1: A partial map of Midtown Manhattan taken from aaccessmaps.

After significant customer complains of slow deliveries, the company is determined to implement a vehicle routing algorithm rather than leave that to its drivers — the way things are currently run. Following UPS' successful strategy, the company decides to follow a similar scheme:

- Strictly no left turns policy: At every intersection, the vehicle may only either proceed straight or make a right turn (U-turns are entirely out of the question here!)
- Minimal right turns policy: The number of right turns made in the entire route must be kept at a minimum and it does not matter what the total distance of the route is

Knowing that you took CS2040S, your manager tasked you to design an algorithm for routing its deliveries. Suppose the following:

- You have access to a black box Single Source Shortest Path (SSSP) algorithm which solves for the shortest paths to *every* vertex v in the graph from a given source vertex u
- You have the grid map of Manhattan's streets, avenues and intersections
- All roads serve two-way traffic
- The delivery vehicle may only travel from intersection to intersection via streets and avenues
- The start points and destinations will always be intersections
- Some intersections may be closed off due to accidents or roadworks from time to time but the destination will always be reachable by some path

Problem 1.a. Given the start point, initial facing direction of the delivery vehicle (North, South, East, West) and the destination, come up with an algorithm that finds the best delivery route according to your company's policy.

Problem 2. EuroTrip 2077

It's the year 2077, and Europe is facing severe traffic congestion, so much so that trade is impacted because the transportation of essential materials and goods in the continent is impeded. As a result, European nations came together and imposed a *odd-even* traffic policy on all roads across Europe. Under this policy, each road is designated as an "even" road or an "odd" road. For individuals, they are permitted to drive on even roads only on even-numbered days of the month and on odd roads only on odd-numbered days of the month. When a road is closed off to individuals, it is fully utilized for trade-related transportation and essential services. The exception for the odd-even policy is the month of December, where it does not apply. This is to facilitate human migration during the holiday season. Manon is a fresh university graduate from France who is planning for her graduation trip. She decides to do a road trip *starting in Paris* and *ending in Moscow*. To plan her trip, she obtained a roadmap consisting of all European cities and the roads connecting them along with the odd-even schedules for every road.

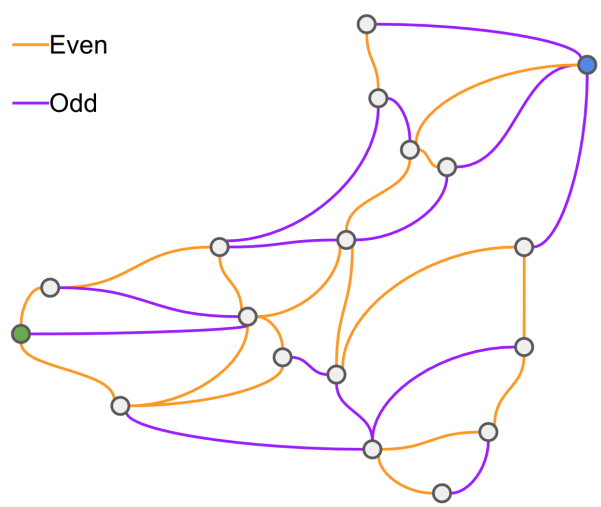

Figure 2: Example of an odd-even roadmap.

For the rest of this problem, assume the following:

- Each road connecting two cities takes about one day to drive (leaving time for a little tourism after Manon arrives)
- All months are 30 days, so odd and even days always alternate!
- There is at least one one valid route to the destination

Problem 2.a. Suppose Manon's trip is planned for the month of December when the odd-even scheme is not in effect (i.e., you can use all the roads every day).

With the goal of finding a driving route from Paris to Moscow that is the *fastest*, model this as a graph problem and solve it using a *single* graph algorithm you have learnt in class so far. Why does the algorithm work in this problem?

Problem 2.b. Suppose now Manon will have to take into account the even-odd scheme because her planned trip will *not* overlap with any days of December. Suppose also that she will have to leave from Paris on an *even* day and she will *not* be spending overnight in any city.

By constructing a new graph, show how to find a driving route from Paris to Moscow that is the *fastest*, using a *single* graph algorithm you have learnt in class so far.

Problem 2.c. What if Manon has a choice of whether to leave on an odd or an even day and also whether or not to stay overnight at a city?