**CS2040S: Data Structures and Algorithms**

# Recitation 7

*Goals:*

- Model problems using graphs
- Transform graphs to utilize existing algorithms as blackboxes
- Identify and formulate shortest paths 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 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. After significant complains of slow deliveries by customers, the company is determined to implement a vehicle routing algorithm rather than leave that to its drivers — the way things are currently run. Knowing that you took CS2040S, your manager tasked you to design a better algorithm for planning delivery routes. Following UPS' successful strategy, the company decides to follow a similar policy:

- 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

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

Notice that whether a turn is left or right depends on the car's direction as it enters the intersection.

One notable characteristic, amongst many others, of Manhattan is how its roads are laid out to form an uniform grid where roads running North-South are known as *avenues* and roads running East-West are known as streets. Indeed, such a city layout is what makes vehicle routing all the more critical for the company's success — one wrong turn could mean a very dissatisfied customer! An example of Manhattan's intersection-filled layout is shown in Figure 1.


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

*Note:* The Single Source Shortest Path (SSSP) problem solves for the shortest paths to *every* vertex $v$ in the graph from a given source vertex $u$. For this question, you may assume SSSP algorithms are available to you as blackboxes.

**Problem 1.a.** You are provided with a square grid map of Manhattan's streets, avenues and intersections. Given the destination intersection and the source intersection from where the vehicle will be setting off along with its initial direction (North, South, East, West), determine the best route for the driver according to your company's policy. The vehicle may only travel from intersection to intersection via streets and avenues.

For the sake of simplicity, you may assume all roads serve two-way traffic (this is no true in reality).

**Problem 2. EuroTrip 2077**

It's the year 2077 and Europe is facing severe traffic congestions, so much so that trade is impacted because the transportation of essential materials and goods in the continent are 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 roadtrip *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.

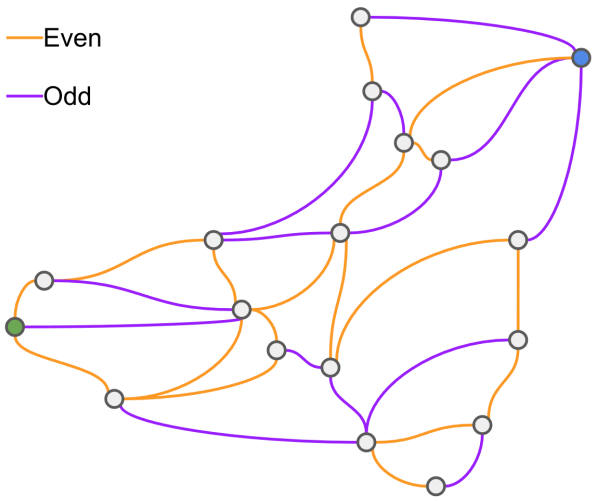


**Figure 2:** Example of an odd-even roadmap.

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 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?

**Problem 3. Single Player Shortest Victory**

Player Unknown Battlegrounds (PUBG) is a popular online multiplayer battle royale game. In the classic solo mode of this game, the winner is the lone survivor. Players in a match are first airdropped into a free-for-all battlefield with no initial inventory. Weapons, ammunition, armours and tools are scattered across the battlefield. Once in a while, special supplies are dropped onto the battlefield at random locations. These special supplies contain powerful armaments and tools which grant players significant advantages in the game. As the game progresses, the play area also shrinks and eventually converge onto a small circle. Players who stay outside the play area suffer consistent damage until their characters perish. This is to force last survivors to come out of hiding and confront each other in a final battle.

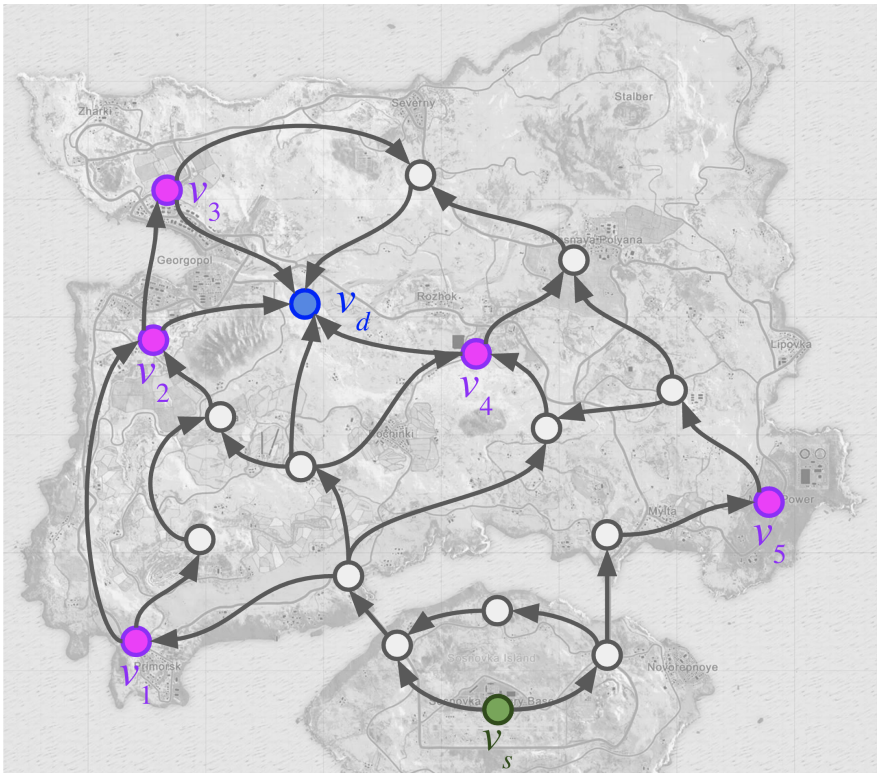


**Figure 3:** Graph of travelling routes in Erangel map.

Suppose you are currently playing a PUBG match on the popular Erangel map. As the match progresses into the final moments, you must strategically plan out your route to the final play area so that you may reach it in the least time. However, you are ill-equipped for the final battle and therefore have to visit at least one of the supply drops along the way. To help yourself find such an optimal route, you put on your CS2040S thinking cap and modelled the map as a graph (Figure 3) like follows:

- A vertex $u$ is a location landmark
- An edge $(u, v)$ represents a route going from location $u$ to $v$

- Assume:
  - Length of edges drawn do not correlate to their travel times
  - Each edge incurs 1 unit of travelling time

**Problem 3.a.** For unweighted directed graphs, how do you compute SSSP on them? What is its time complexity?

**Problem 3.b.** With reference to Figure 3, you have a *directed weighted* graph $G = (V, E)$ where $V$ is the set of vertices and $E$ is the set of edges. Your current location is at $v_s$ (green vertex) and the final play area is converging onto $v_d$ (blue vertex). There are $v_1, v_2, \ldots, v_k$ supply drops scattered on the map (magenta vertices).

How do you find find the *fastest* route starting from $v_s$ and ending at $v_d$ such that you visit *at least* one supply drop $v_i \in \{v_1, \ldots, v_k\}$ along the way?