CS2040S: Data Structures and Algorithms

Recitation 8

Goals:

- Advanced graph modelling
- Apply the idea of triangle inequality in shortest paths
- Explore applications of shortest paths problems

Problem 1. StonksX Trader

One day you decided to get into the Foreign Exchange Market (Forex). You registered as a trader in the *StonksX* exchange, a popular exchange for global currencies. There are currently n currencies being traded in the exchange. To help you with analysis, you created a matrix R containing all exchange rates where $R[i, j]$ is the amount of currency j you can get for one unit of currency i. Note that exchange rates are not symmetric: $R[i, j] \neq R[j, i]$. Alas, there is "no arbitrage" possible in the *StonksX* exchange, meaning that if you start with one currency and then convert it to another, and then another, and so on, and then back to the first currency, you will end up with *no more* money then what you started with.

You begin the trading day with an account full of Singapore Dollars (SGD), and you want to end the day with only Great Britain Pounds (GBP). Find the sequence of conversions that will maximize the amount of GBPs you have at the end of the trading day.

Problem 1.a. How do you model this problem as a graph? What are its vertices and edges? Is it weighted or non-weighted? Are the edges directed or non-directed?

Problem 1.b. In your graphical model, which of its property reflects the "no arbitrage" rule? What would happen if such a rule is not enforced by the exchange?

Problem 1.c. What graph algorithm will you use to solve the problem? Which modifications are necessary?

Challenge question: How can you use the algorithm without modifying it (i.e. in the form that was presented to you in lecture)?

Problem 2. 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.

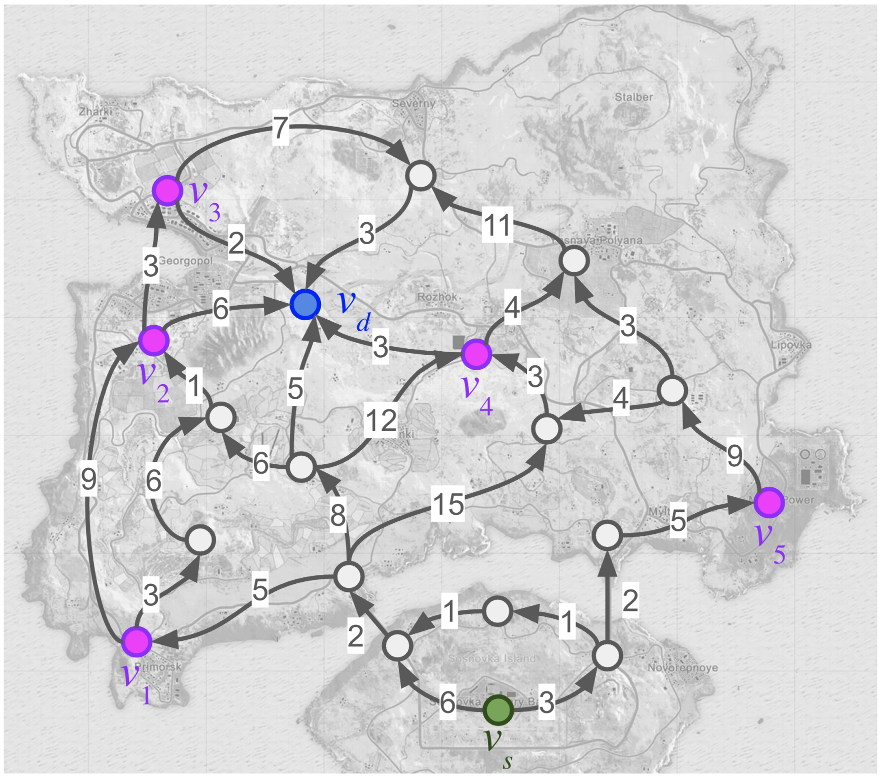


Figure 1: Graph of travelling routes on Erangel map where numbers represent time needed.

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 in the following manner:

- 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

With that, you drew up a *directed weighted* graph as seen in Figure 1. Let us denote the 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).

Problem 2.a. Come up with an algorithm to find the *quickest* route starting from v_s and ending at v_d such that you visit *at least* one supply drop $v_i \in \{v_1, \ldots, v_k\}$. State the space and time complexities as well as the limitations (if any) of your solution.

Problem 3. [Optional Question] Winning the Games [1]

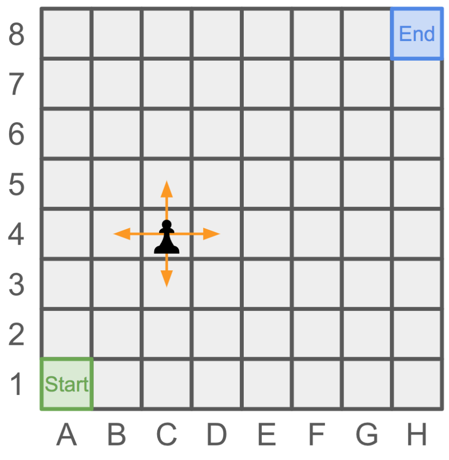
For this problem, consider an $n \times n$ chessboard consisting of n columns and n rows. We have labelled the columns with letters and the rows with numbers. For instance in Figure 2a, your piece is at position C4.

Every square on the board has a reward associated with it, and your goal is to move a piece from the bottom left corner to the top right corner, while collecting as much points as possible.

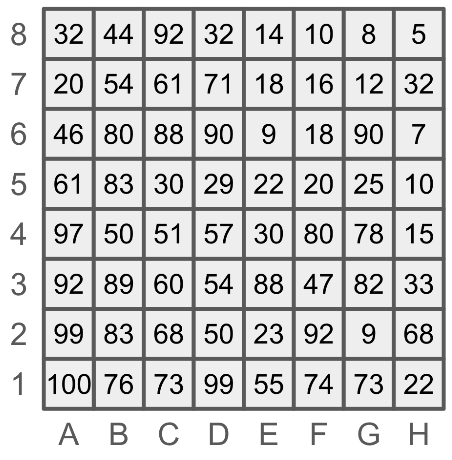
The rules of the game are as follows:

- In every move, your piece can move one square in either the vertical or horizontal direction. It cannot move diagonally. For example, in Figure 2a, the piece at square C4 can potentially move to the squares: C5, C3, D4, B4. It cannot move to square D5, D3, B5, or B3.
- Your piece can only move to a square of lesser value. That is, if your piece is on a square with reward r, it can only move to a square with reward $< r$. In Figure 2b above, the piece at square C4 can move to square C5 and B4 (since $30 < 51$ and $50 < 51$), but not to any other square.
- Your piece starts in square A1.

[1] Optional questions will not be covered during the actual recitation



(a)



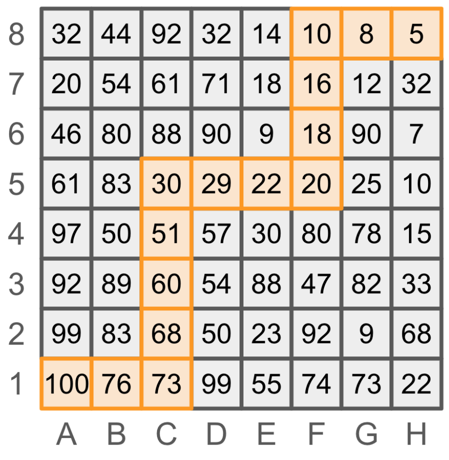
	A	B	C	D	E	F	G	H
8	32	44	92	32	14	10	8	5
7	20	54	61	71	18	16	12	32
6	46	80	88	90	9	18	90	7
5	61	83	30	29	22	20	25	10
4	97	50	51	57	30	80	78	15
3	92	89	60	54	88	47	82	33
2	99	83	68	50	23	92	9	68
1	100	76	73	99	55	74	73	22

(b)

Figure 2

- When your piece reaches the square in the top right corner of the board (in the example above, H8), it is rewarded with the value of every square it traversed.

Figure 3 below shows an example of a legal set of moves (with a reward of 586).



	A	B	C	D	E	F	G	H
8	32	44	92	32	14	10	8	5
7	20	54	61	71	18	16	12	32
6	46	80	88	90	9	18	90	7
5	61	83	30	29	22	20	25	10
4	97	50	51	57	30	80	78	15
3	92	89	60	54	88	47	82	33
2	99	83	68	50	23	92	9	68
1	100	76	73	99	55	74	73	22

Figure 3

Problem 3.a. Explain how the problem can be modelled as a directed graph. How many nodes does your graph have? State one important property of your graph. Draw a small example (e.g., for $n = 2$ or $n = 3$) to illustrate.

Problem 3.b. Give an efficient algorithm for finding the sequence of moves which attains the maximum possible reward. You may use any algorithm from class in unmodified form without reproducing it. What is the (asymptotic) performance of your algorithm?