CS2040S: Data Structures and Algorithms

Recitation 8

Goals:
- Advanced graph modelling
- Apply the idea of triangle inequality in shortest paths
- Explore shortest paths applications in MED and constraint problems

Problem 1. Minimum Edit Distance (MED)

We have already seen the problem of comparing two strings several times in this class. Another metric for measuring the difference between two strings is their *edit distance*: what are the character edits necessary to transform one string to the other. For a given string S, the character edit operations that we measure edit distance with are as follows:

Operation	**Behaviour**
insert(pos, c)	Inserts a character c at position pos in string S.
delete(pos)	Removes character at position pos in string S.
replace(pos, c)	Replace a character at position pos in string S with character c.

For example, suppose you are given the following two strings representing nucleotide sequences:

S_1: AGGAACCGTA
S_2: AGAATCCGA

One valid sequence of edits to transform from S_1 to S_2 is as follows:
1. delete(2): Delete G at position 2
2. delete(7): Delete T at position 7
3. insert(4, T): Insert T at position 4

Problem 1.a. Given string u of length m and v of length n, how can we use a *directed graph* to capture all possible sequences of edits to transform u to v?

Show the sequences for transforming the string "CG" to the string "AGT" by illustrating the graph in drawing.

Problem 1.b. Our goal here is to obtain the *minimum edit distance* (MED) between 2 strings. MED is calculated as the sequence of edits that incurs the *least total cost* out of all possible sequences. Suppose that an insert has cost 3, a delete has cost 5, and a replace has cost 7. How would you find the MED to transform one string into the other? What graph problem is this?

Problem 2. Constraint Scheduling

Airport runway scheduling is a challenging problem for air traffic controllers. Optimizing runway scheduling for takeoff and landing would in turn optimize for flight throughput at an airport. Today, we shall look at scheduling aircraft takeoffs at an airport. One of the main challenges of takeoff scheduling is the physical phenomenon of wake turbulence caused by an aircraft in flight which creates a rotating mass of air. For safety reasons, the presence of such wake vortices necessitates time separation between successive takeoffs. The duration of these separations is often calculated based upon the relative sizes of the leading and trailing aircraft. For instance, it would be dangerous for a small jet to takeoff immediately after a large cargo plane left the runway. At the same time, to facilitate a smooth air traffic flow, aircrafts taking the same runway and flightpath should be scheduled as closely together as possible. This means that the time separation between some successive takeoffs should be bounded within a reasonable time frame and not be too far apart from each other.

You are provided a set of aircraft takeoffs $t_1, t_2, \ldots, t_n$ along with their accompanying time dependency constraints. An example for scheduling takeoffs t_1, t_2, t_3, t_4 is given as follows:

Takeoff	**Time dependencies**
t_1	• *at least* 1 minute before t_3
t_2	• *at least* 6 minutes before t_1 • *at least* 2 minutes before t_3
t_4	• *at least* 1 minute before t_2 • *at most* 8 minutes before t_3 • t_4 and t_1 *at most* 7 minutes *apart* from each other

Your goal here is to determine the *feasibility* of this collection of takeoffs and their time dependency constraints, and if it is, find such an optimal schedule. Optimal here means that the time taken to fulfill all takeoffs in the schedule is the minimum out of all possible schedules. In the given example, the optimal schedule is achievable in 8 minutes: $t_3 \xrightarrow[\text{minute}]{1} t_1 \xrightarrow[\text{minutes}]{6} t_2 \xrightarrow[\text{minute}]{1} t_4$.

Problem 2.a. Model this problem as a graph and cast it as a shortest paths problem. *Hint:* For an edge to t_i, think of its weight as the *relative* time at which takeoff t_i is scheduled. Don't worry about negative times/weights because time is relative!

Problem 2.b. How do you use this graph to find a schedule that satisfies the constraints? What does it mean if there is a negative weight cycle?