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

We have already seen the problem of comparing two strings several times in this module. Another metric for measuring the difference (or equivalently, similarity) between two strings is their *edit distance*. Edit distance measures the *total cost* of transforming from one string to the other via a sequence of character edit operations. In the typical version of this problem, there are three permitted character edit operations:

Operation	**Behaviour**	**Cost**
insert(i, c)	Inserts a character c at position i in the string.	$\$ins$
delete(i)	Removes character at position i in the string.	$\$del$
replace(i, c)	Replace a character at position i in the string with character c.	$\$rep$

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

S: `AGGAACCGTA`
T: `AGAATCCGA`

One valid sequence of edits to transform from source string S to target string T is as follows:

1. delete(2): Delete `G` at position 2
2. delete(7): Delete `T` at position 7
3. insert$(4, \texttt{T})$: Insert `T` at position 4

If every edit operation has a cost of 1 (i.e. $\$ins = \$del = \$rep = 1$), then the edit distance due to the sequence of edits above is 1+1+1=3.

Clearly, many possible edit sequences are possible and therefore we are only interested in the *Minimum Edit Distance* (MED): The minimum out of all possible edit distances for transforming a source string to a target string.

Note that the cost for each edit operation varies from problem to problem. When we change the edit operation costs, we may end up with a different MED corresponding to a different sequence of edits.

Problem 1.a. How can we represent the MED between 2 strings as a graph? Let m be the length of source string S and n be the length of target string T, how many vertices and edges does your MED graph have (in terms of m and n)? Draw your MED graph for transforming the string `CG` to the string `AGT`.

Problem 1.b. Suppose that $\$ins = 3$, $\$del = 5$ and $\$rep = 7$. How would you compute the MED between two strings? What graph problem is this? What is the time complexity of your algorithm? Illustrate your solution using the same example of transforming the string `CG` to the string `AGT`.

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: $t_4 \xrightarrow[\text{minute}]{1} t_2 \xrightarrow[\text{minutes}]{6} t_1 \xrightarrow[\text{minute}]{1} t_3$.

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?