CS 341: Algorithms
Module 7: Graph Algorithms

Armin Jamshidpey, Eugene Zima

Based on lecture notes by many previous CS 341 instructors

David R. Cheriton School of Computer Science, University of Waterloo

Spring 2019

Formulating Problems as Graph Problems: 1

- Reliable network routing:
  - Suppose we have a computer network with many links.
  - Every link has an assigned reliability.
    - ★ The reliability is a probability between 0 and 1 that the link will operate correctly.
  - Given nodes u and v, we want to choose a route between nodes u and v with the highest reliability.
    - ★ The reliability of a route is a product of the reliabilities of all its links.

Solution:

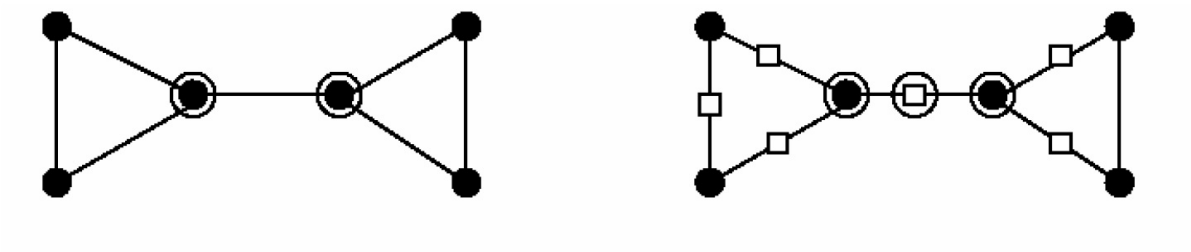
- The route will correspond to a path in the graph.
- Can we make this look like a shortest path problem?
- Yes:
  - Since reliability is computed as a product, we will want to change the weights so that an edge is assigned the logarithm of the probability.
    - ★ Then we sum logs to work with products of probabilities.
  - To get the best reliability path we want the highest probability of operation which we can derive by finding the least weight path if the assigned weights are negative logarithms of the probability values.
    - ★ Then we are able to use Dijkstra's algorithm.

Formulating Problems as Graph Problems: 2

Bridges in Graphs:

- Suppose we have a computer network with many links.
  - We can assume the network is currently connected so as to enable communication between any two nodes of the network.
- We want to identify the critical network links.
  - A link is critical (also called a bridge) if its removal (due to a communication malfunction) causes a lack of communication between some pair of nodes in the network.
- Hint:
  - Should we find all edges between two articulation points?
  - You should determine why this is a bad strategy.
    - ⋆ But perhaps there is still some way to use articulation points ...

- A different approach:
  - We view both the network nodes and network links as nodes in our graph representation.
  - We connect a link-vertex to a node-vertex if the network link has an endpoint in the network node.
  - Then, a link is critical (i.e. a bridge) if and only if the corresponding link-vertex is an articulation point.



Formulating Problems as Graph Problems: 3

The Greyhound bus problem:

- Suppose we are given a bus schedule with information for several buses. A bus is characterized by four attributes:
  - the "from-city", the "to-city", departure time, arrival time.
- What buses should be taken to go from city F to city T if we want the fastest trip?
  - We want to have wait times between buses taken into account.
- First, let's eliminate an idea leading to an inadequate solution:
  - We could use a graph that has nodes representing cities.
  - Label each edge with the travel time between cities.
  - Now go for the least cost path.
    - ★ BUT: there is no accounting for wait times!
    - ★ Also, there may be several different travel times between two adjacent cities.
  - But there is another way to use a graph strategy ...

Sample Bus Schedule

UW to Hamilton	15:40 17:25		
UW to Toronto	09:00 11:00	17:00 19:00	
Hamilton to Niagara Falls	17:30 18:45		
Toronto to Niagara Falls	12:30 14:05	20:30 22:10	
Niagara Falls to Buffalo	14:10 15:25	18:40 19:40	22:55 23:59

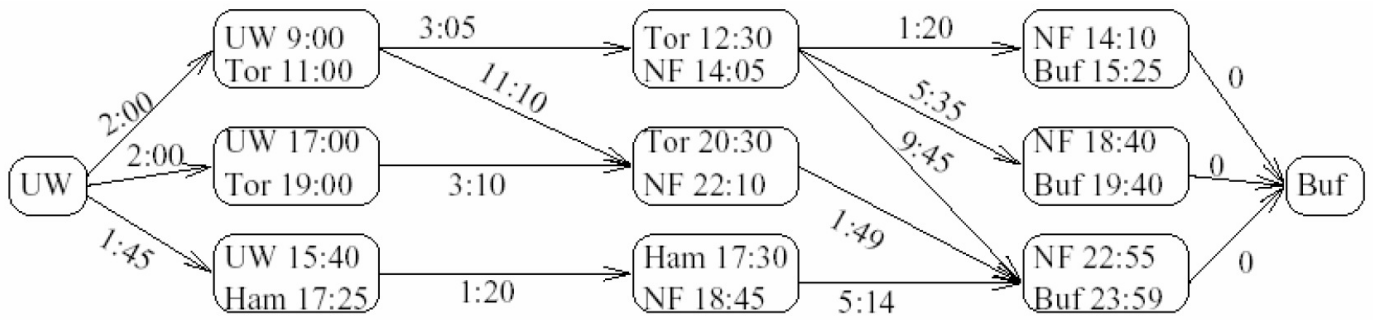
- Another approach:
  - Use a graph in which each vertex is a bus.
  - There will be an edge between buses x and y if and only if:

$$x.\texttt{to\_city} = y.\texttt{from\_city} \quad \textit{and}$$
$$y.\texttt{departure\_time} \geq x.\texttt{arrival\_time}.$$

  - Our time cost for an edge will be:

$$\begin{aligned} & \texttt{waiting\_time} + \texttt{travel\_time on bus} \\ y = & (y.\texttt{departure\_time} - x.\texttt{arrival\_time}) + \\ & (y.\texttt{arrival\_time} - y.\texttt{departure\_time}) \\ = & \, y.\texttt{arrival\_time} - x.\texttt{arrival\_time}. \end{aligned}$$

- We also need two special vertices for the origin and destination cities.
  - There is an edge from the origin to bus x if and only if $x.\texttt{from\_city} = \textit{origin}$.
  - The time cost of this edge is $x.\texttt{arrival\_time} - x.\texttt{departure\_time}$.
  - There is an edge from bus y to the destination if and only if $y.\texttt{to\_city} = \textit{destination}$.
  - The time cost of this edge is 0.
- We now have a shortest path problem:

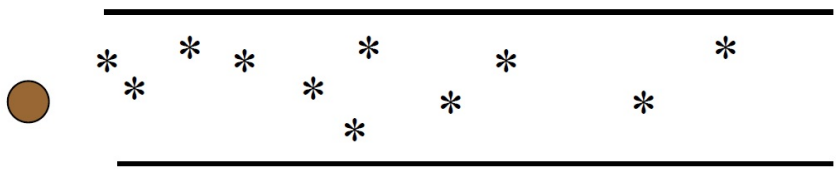


  - Note: the shortest trip is via Toronto with time 6 : 25 hours.

Formulating Problems as Graph Problems: 4

The RootBear Problem:

- Suppose we have a canyon with perpendicular walls on either side of a forest.
  - We assume a north wall and a south wall.
- Viewed from above we see the A&W RootBear attempting to get through the canyon.
  - We assume trees are represented by points.
  - We assume the bear is a circle of given diameter d.
  - We are given a list of coordinates for the trees.
- Find an algorithm that determines whether the bear can get through the forest.



The graph formulation for this problem

- Create a vertex for each tree, and for each canyon wall.
- Two trees are connected by an edge if and only if the RootBear cannot pass between them.
  - That is if their separation is less than d.
  - Do the same for a tree and its perpendicular distance to a canyon wall.
  - Now we determine whether the two canyon walls are in the same connected component of the graph.
    - ★ If they are then the bear cannot pass through the canyon.
    - ★ Otherwise the boundary of the connected component containing the northern canyon wall defines a viable path for the bear.

Conclusion

- Graphs are a very important formalism in computer science.
- Efficient algorithms are available for many important problems:
  - exploration,
  - shortest paths,
  - minimum spanning trees, etc.
- If we formulate a problem as a graph problem, chances are that an efficient non-trivial algorithm for solving the problem is known.
- Some problems have a natural graph formulation.
  - For others we need to choose a less intuitive graph formulation.
  - Some problems that do not seem to be graph problems at all can be formulated as such.