Creating a Road Network Graph

Course 4, Module 3, Lesson 1

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

Learning Objectives

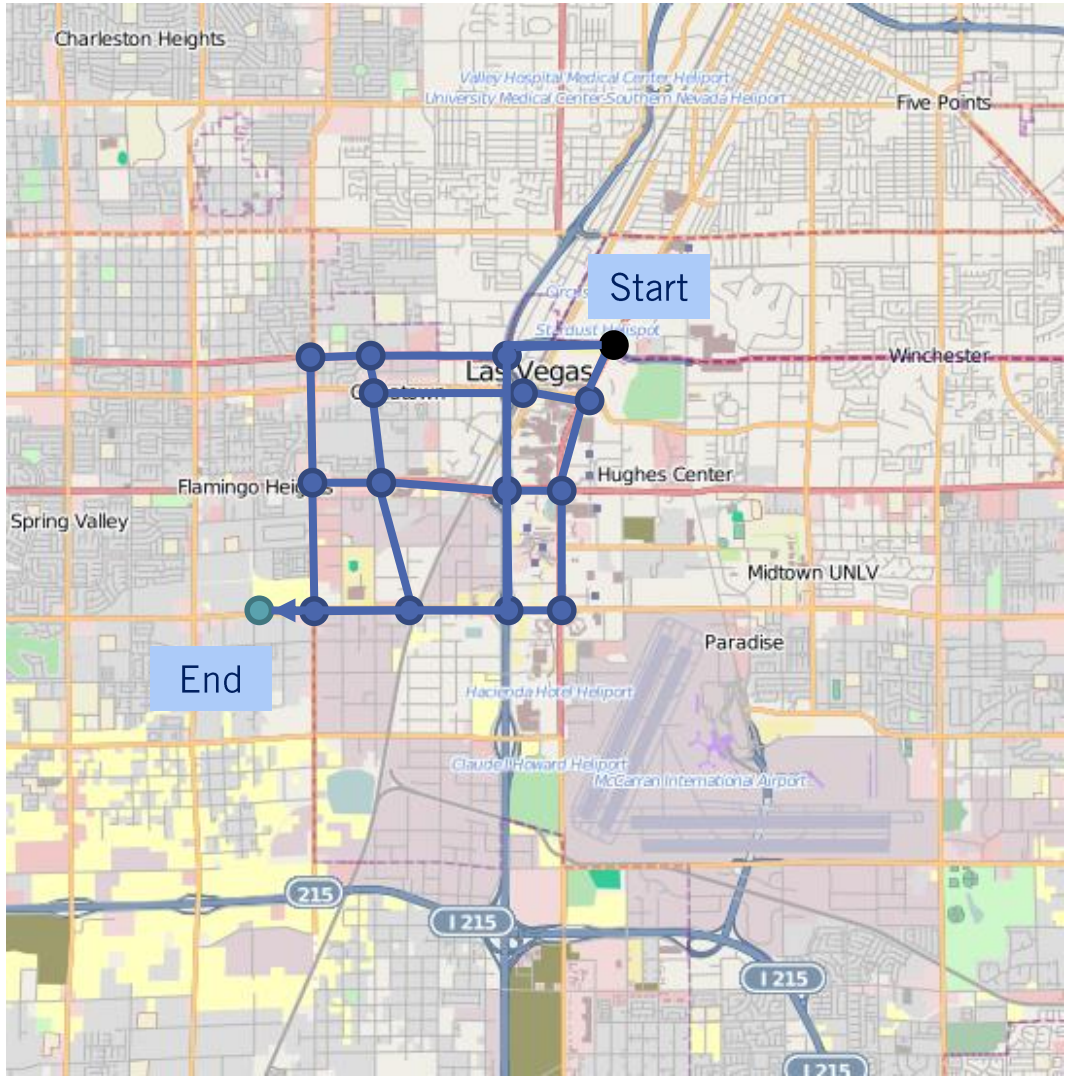
- Understand the mathematical concept of a graph
- Use a directed graph to represent a road network
- Implement Breadth-First Search

Mission Planning

Highest level planning problem.

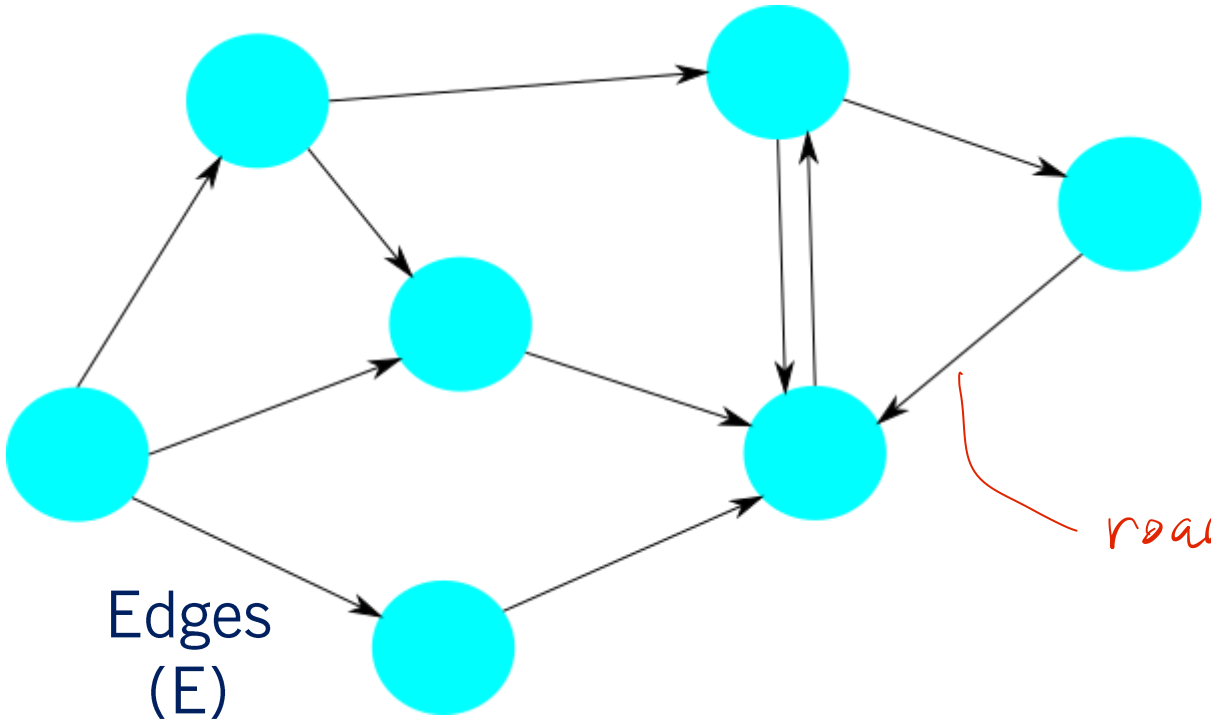
abstracting away lower-level details. Like rules of the road and other agents present

- Order of km.
- Focus on speed limit, road length, traffic flow rate, & road closures



Graphs

Graph: $G = (V, E)$

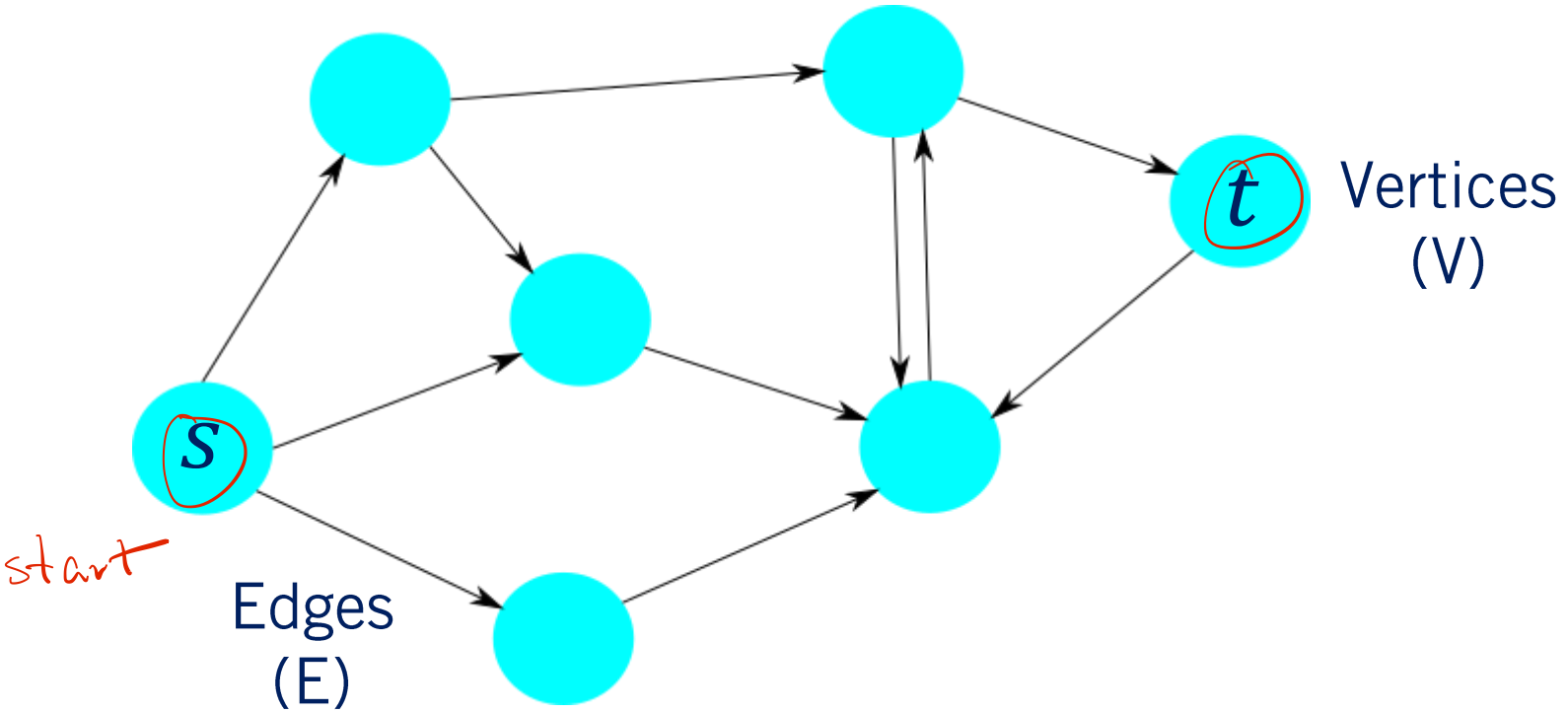


point on the road network

road segment

Graphs

Graph: $G = (V, E)$



Breadth First Search (BFS)

Algorithm BFS(G,s,t)

```
1.  open ← Queue()
2.  closed ← Set()
3.  predecessors ← Dict()
4.  open.enqueue(s)
5.  while !open.isEmpty() do
6.      u ← open.dequeue()
7.      if isGoal(u) then
8.          return extractPath(u, predecessors)
9.      for all v ∈ u.successors()
10.         if v ∈ closed or v ∈ open then
11.             continue
12.         open.enqueue(v)
13.         predecessors[v] ← u
14.     closed.add(u)
```

First in First out

Example - First Wavefront

Open Queue:

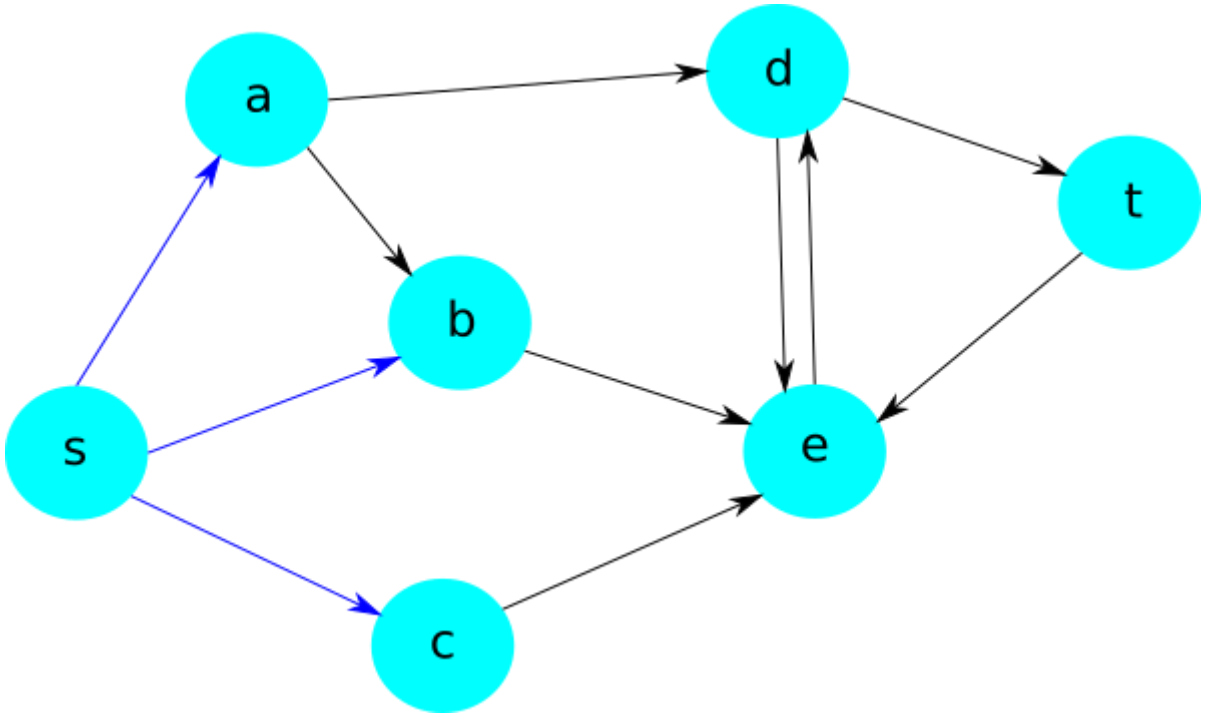
a

b

c

Closed Set: s

Predecessors: S



Example - Second Wavefront

Open Queue:

d

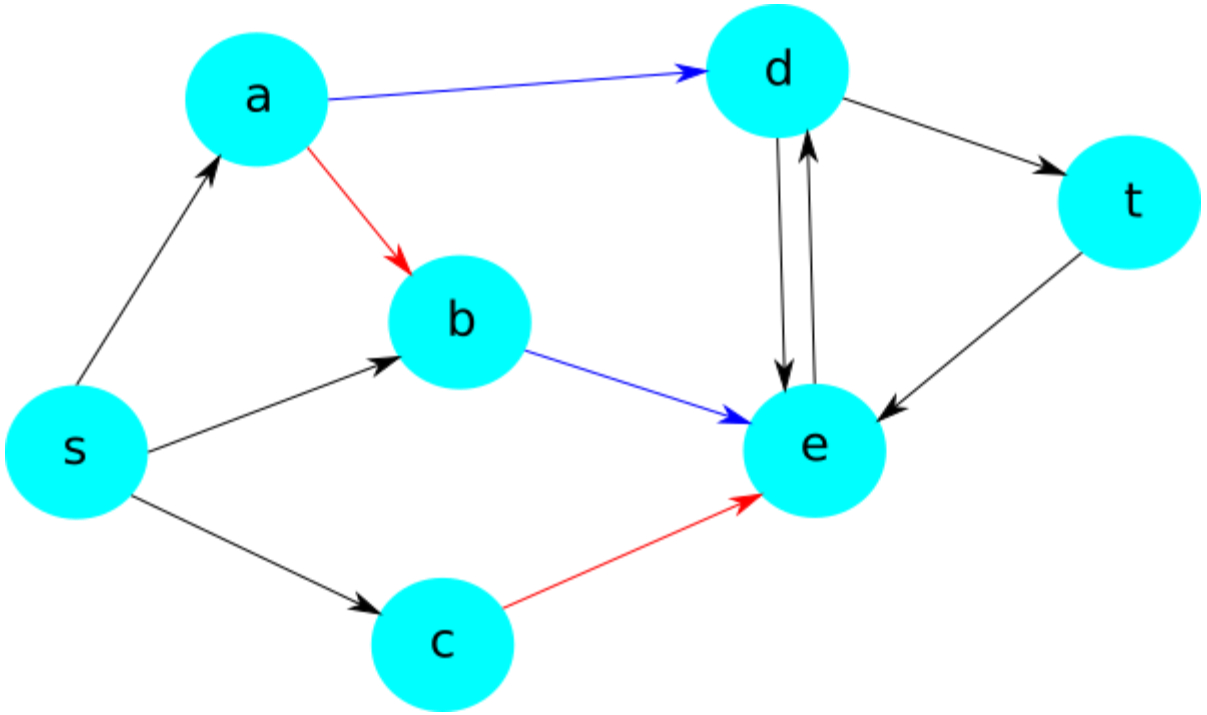
e

Closed Set: s

a

b

c



Example - Third Wavefront

Open Queue:

t

Closed Set: s

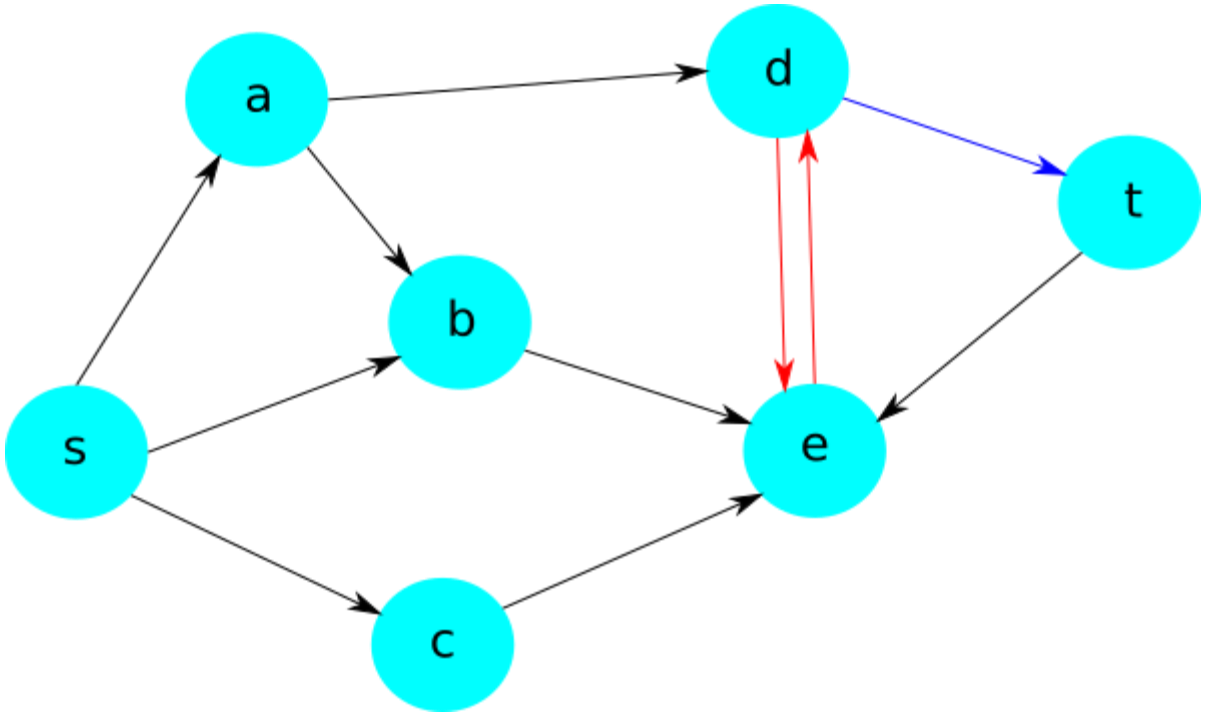
a

b

c

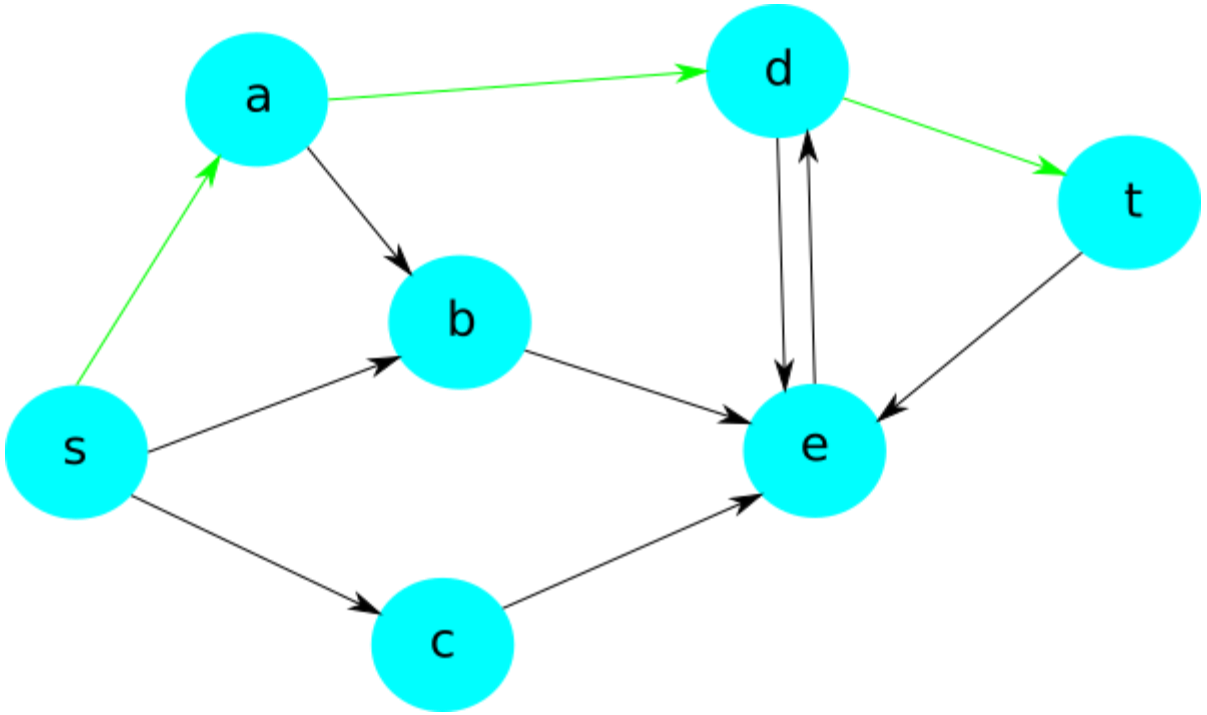
d

e



Example - Optimal Path

Final Path: s
a
d
t



Summary

- Recognize the mission planning problem as a map-level navigation problem
- Learned how to embed a graph in the map
  - Vertices connected by road segments, which correspond to edges
- Learned how to use BFS to search an unweighted graph for the shortest path to the destination

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING