A* Shortest Path Search

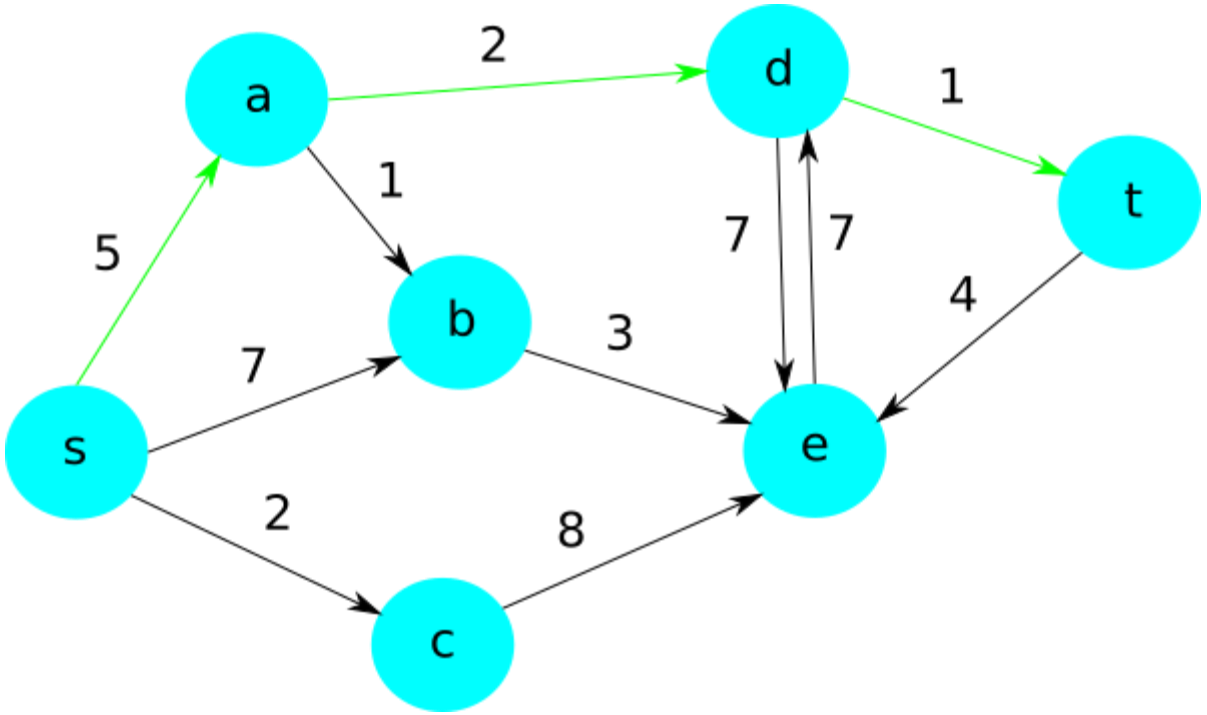
Course 4, Module 3, Lesson 3

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

Learning Objectives

- Understand what admissible heuristics are in the context of graph search
- Understand how to use the Euclidean heuristic to improve our mission planning speed in practice
- Implement the A* search algorithm, leveraging the Euclidean heuristic
- Understand how to apply A* search to variants on the mission planning problem involving time instead of distance

Recall: Dijkstra's for Weighted Graph



Euclidean Heuristic

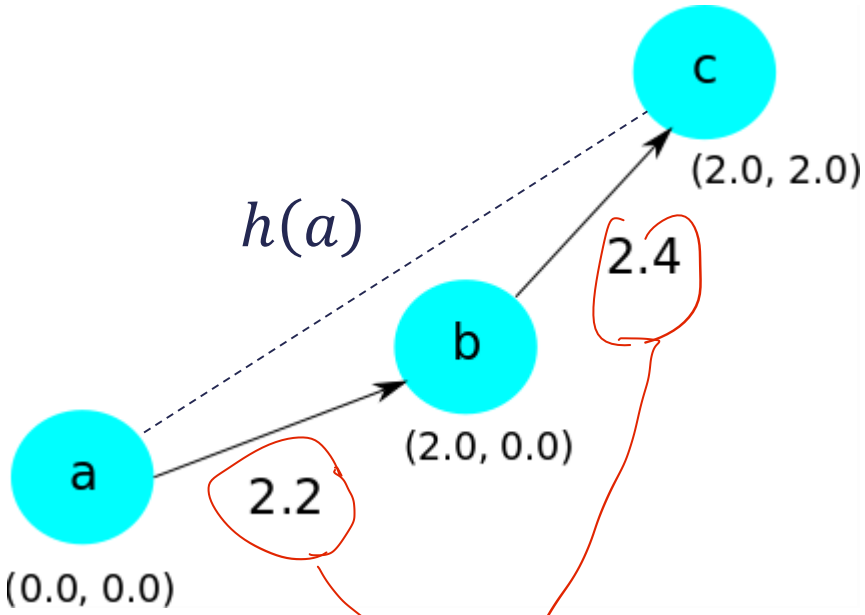
- Exploits structure of the problem
- Fast to calculate
- Straight-line distance between two vertices is a useful estimate of true distance along the graph

$$h(v) = \|t - v\|$$

Euclidean Heuristic - Example

Admissible heuristic
(h ≤ true cost)

$$h(a) = \sqrt{2^2 + 2^2} = 2.828$$



road segments are not straight line paths

A* Algorithm

use heuristic to guide the search to the goal

A* Algorithm

Algorithm A*(G,s,t)

```
1.  open ← MinHeap()
2.  closed ← Set()
3.  predecessors ← Dict()
4.  open.push(s, 0)
5.  while !open.isEmpty() do
6.    u, uCost ← open.pop()
7.    if isGoal(u) then
8.      return extractPath(u, predecessors)
9.    for all v ∈ u.successors()
10.     if v ∈ closed then
11.       continue
12.   uvCost ← edgeCost(G, u, v)
13.   if v ∈ open then
14.     if uCost + uvCost + h(v) < open[v] then
15.       open[v] ← uCost + uvCost + h(v)
16.       costs[v] ← uCost + uvCost
17.       predecessors[v] ← u
18.   else
19.     open.push(v, uCost + uvCost)
20.     costs[v] ← uCost + uvCost
21.     predecessors[v] ← u
22.   closed.add(u)
```

A* Algorithm

accumulated cost from start

heuristic estimate to the goal

total cost $g = f + h$

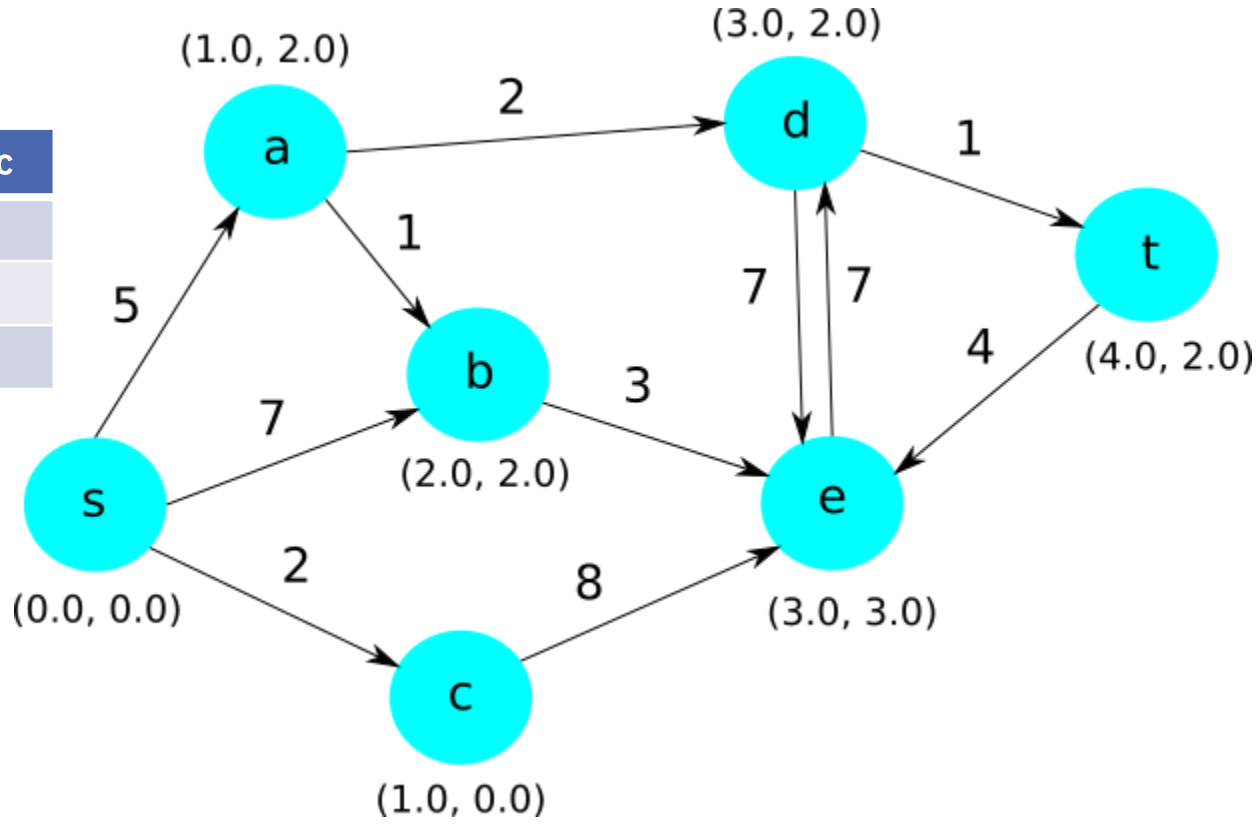
1. **if** $v \in$ open **then**
2. **if** $uCost + uvCost + h(v) <$ open$[v]$ **then**
3. open$[v] \leftarrow uCost + uvCost + h(v)$
4. costs[v] $\leftarrow uCost + uvCost$
5. predecessors$[v] \leftarrow u$
6. else
7. open.push$(v, uCost + uvCost)$
8. costs[v] $\leftarrow uCost + uvCost$
9. predecessors$[v] \leftarrow u$

Example - Origin Node

Open Min Heap:

Node	Cost + Heuristic
s	4.472

Closed Set:

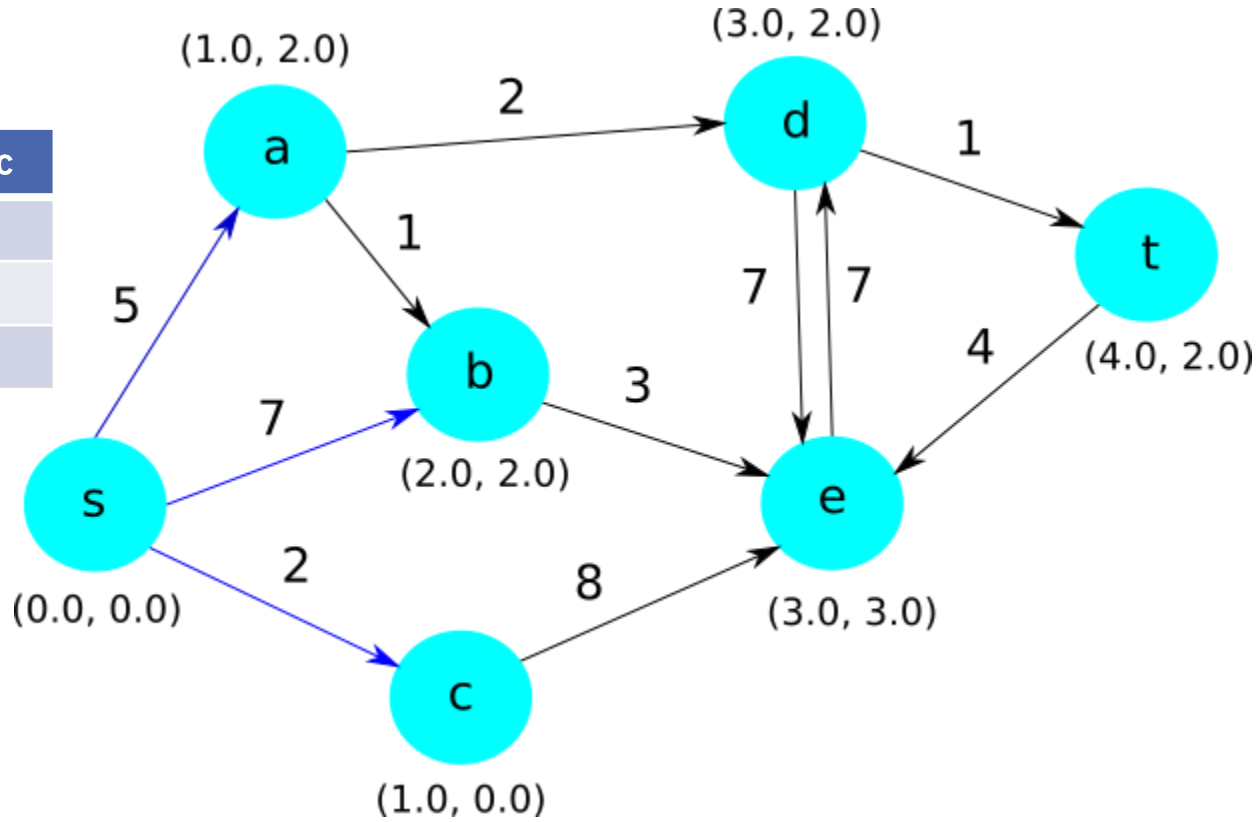


Example - Processing s

Open Min Heap:

Node	Cost + Heuristic
c	5.606
a	8
b	9

Closed Set: s



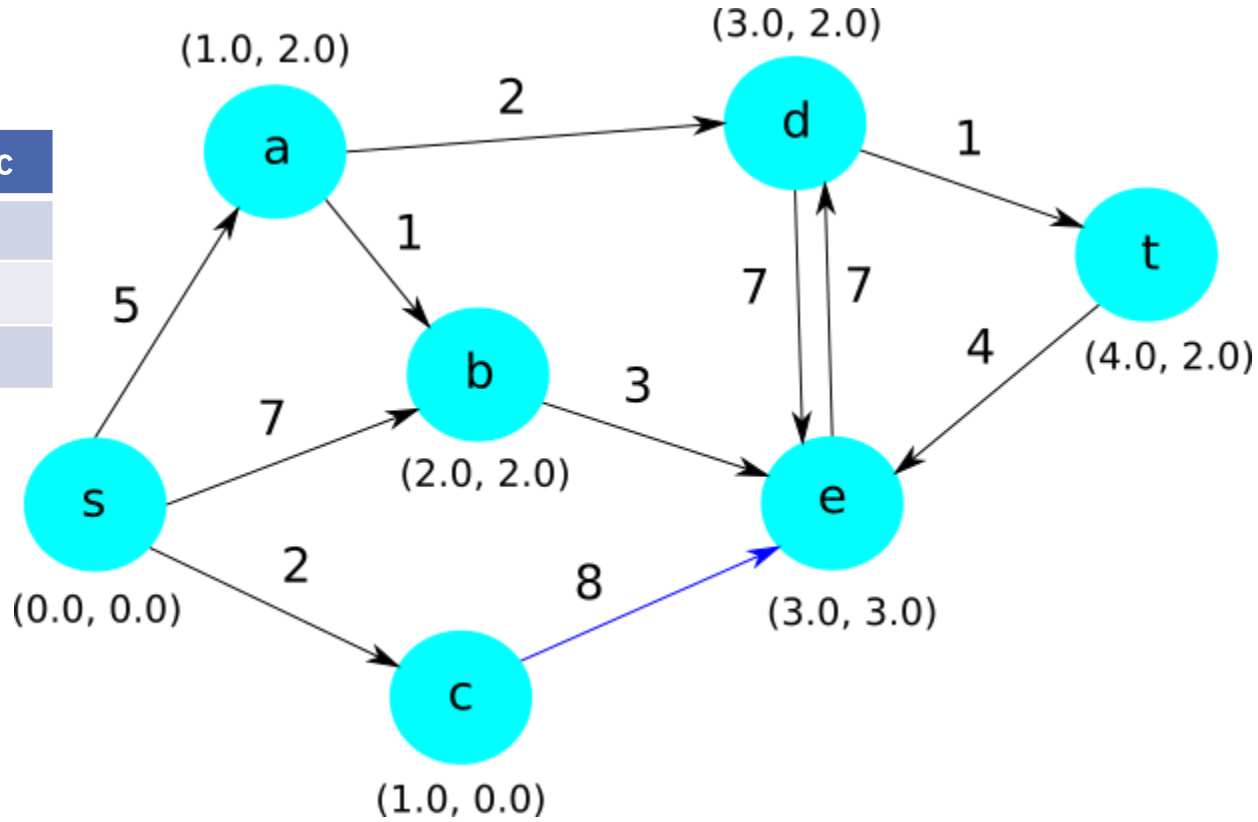
Example - Processing c

Open Min Heap:

Node	Cost + Heuristic
a	8
b	9
e	11.414

Closed Set: s

c

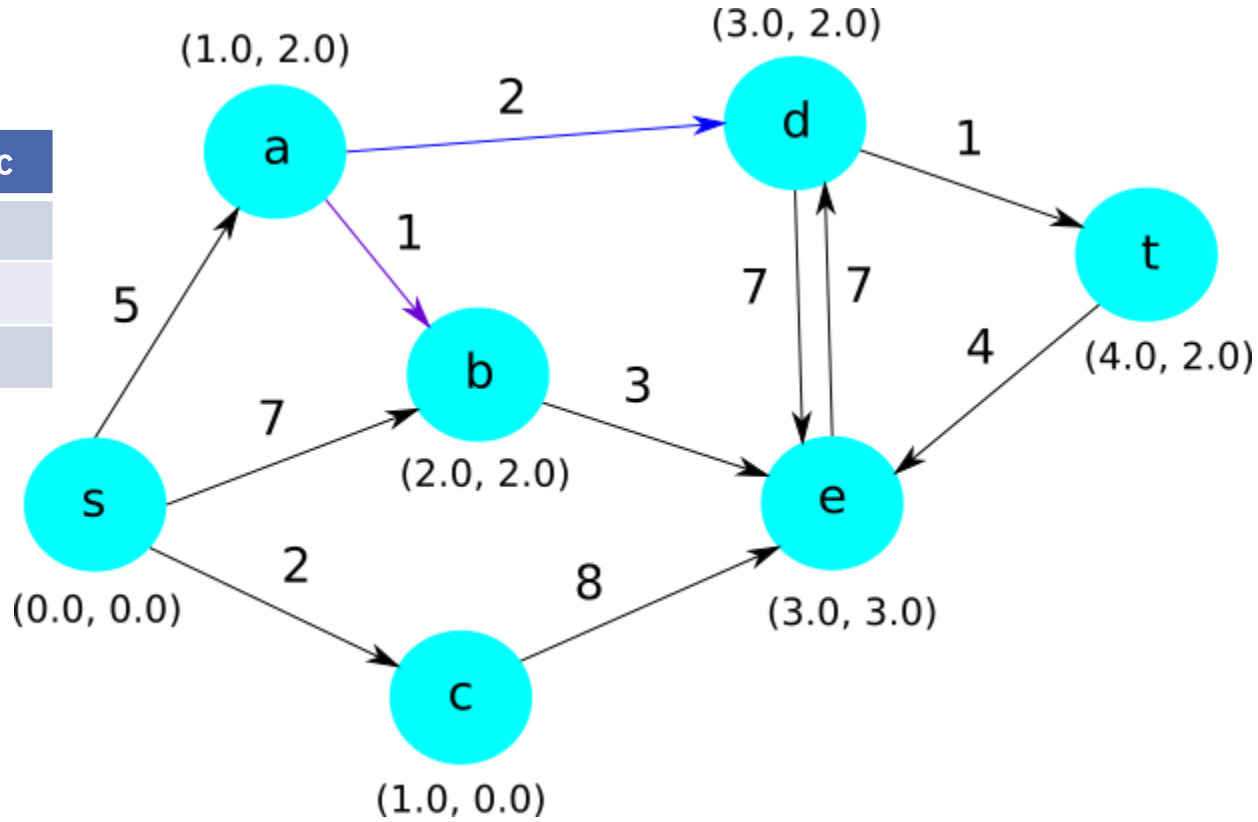


Example - Processing a

Open Min Heap:

Node	Cost + Heuristic
d	7
b	8
e	11.414

Closed Set: s
c
a

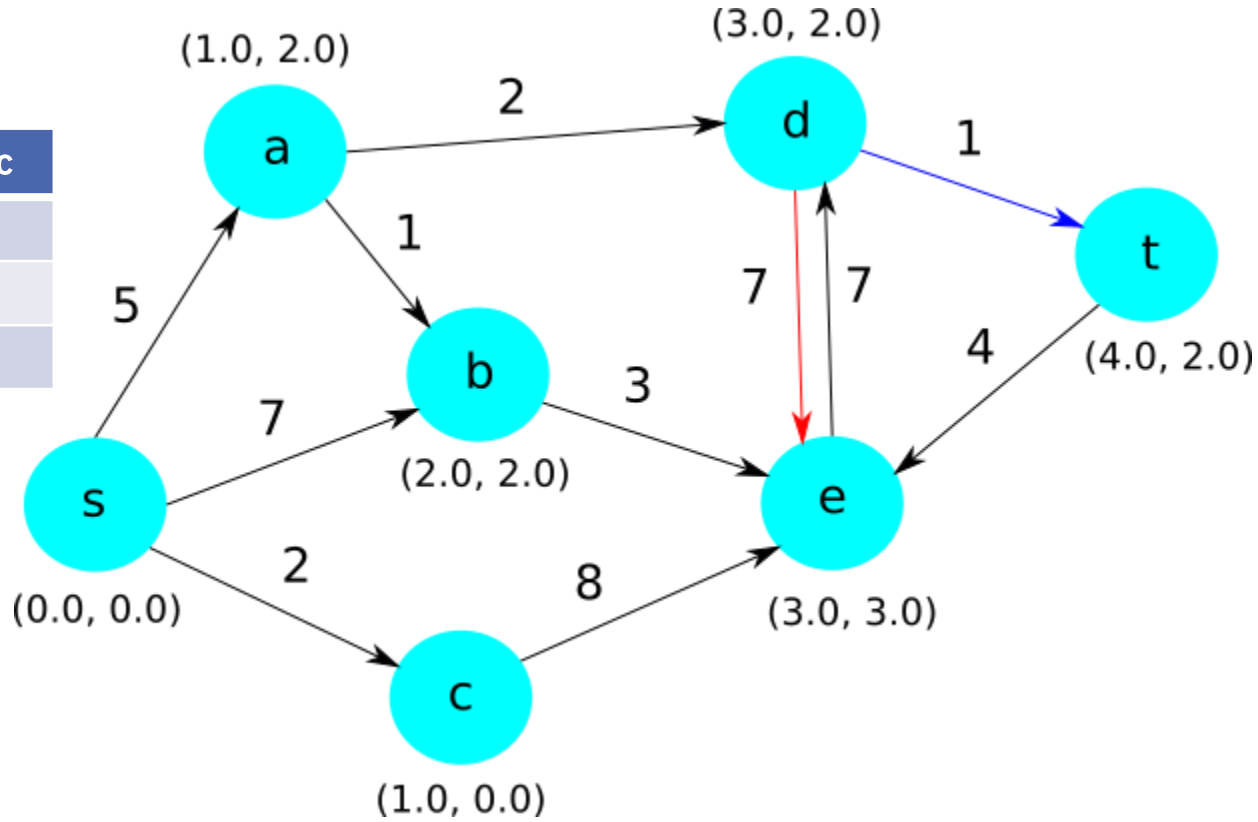


Example - Processing d

Open Min Heap:

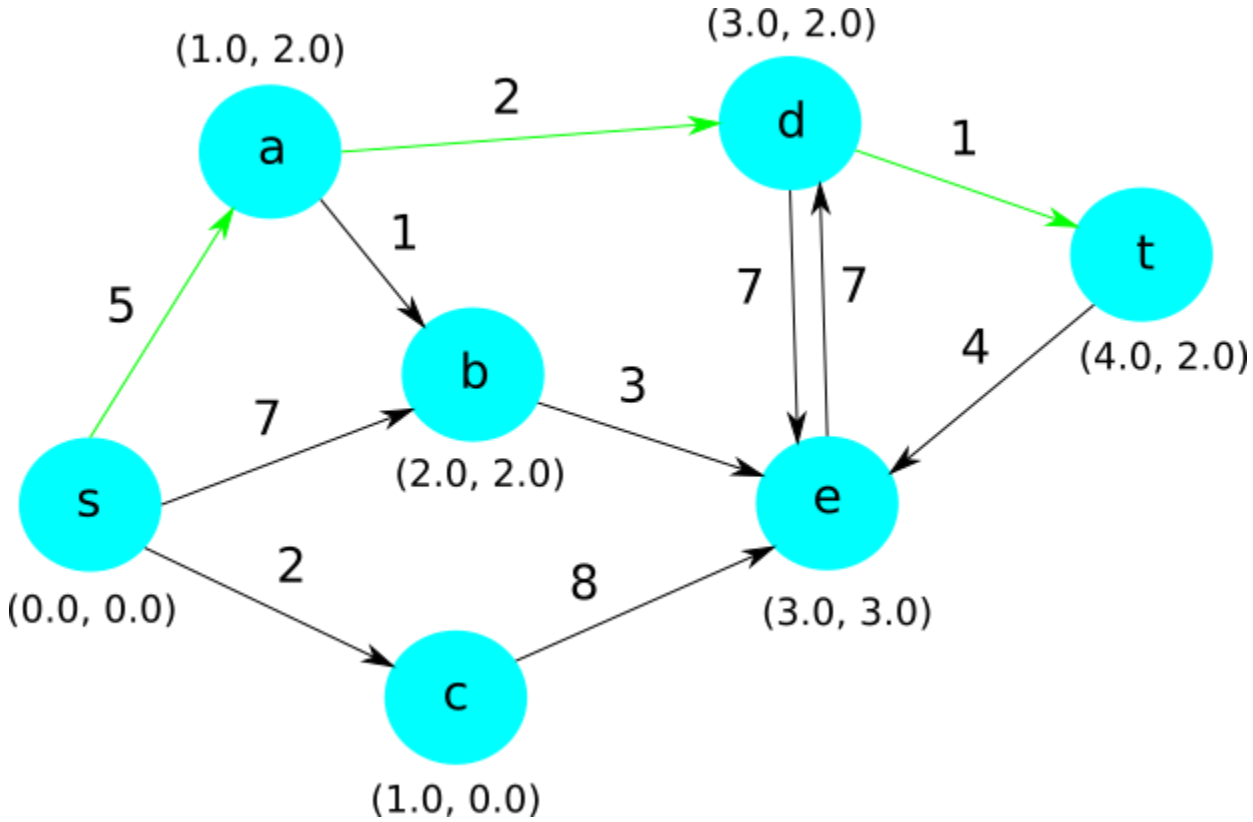
Node	Cost + Heuristic
t	7
b	8
e	11.414

Closed Set: s
c
a
d



Example - Final Path

Final Path: s
a
d
t

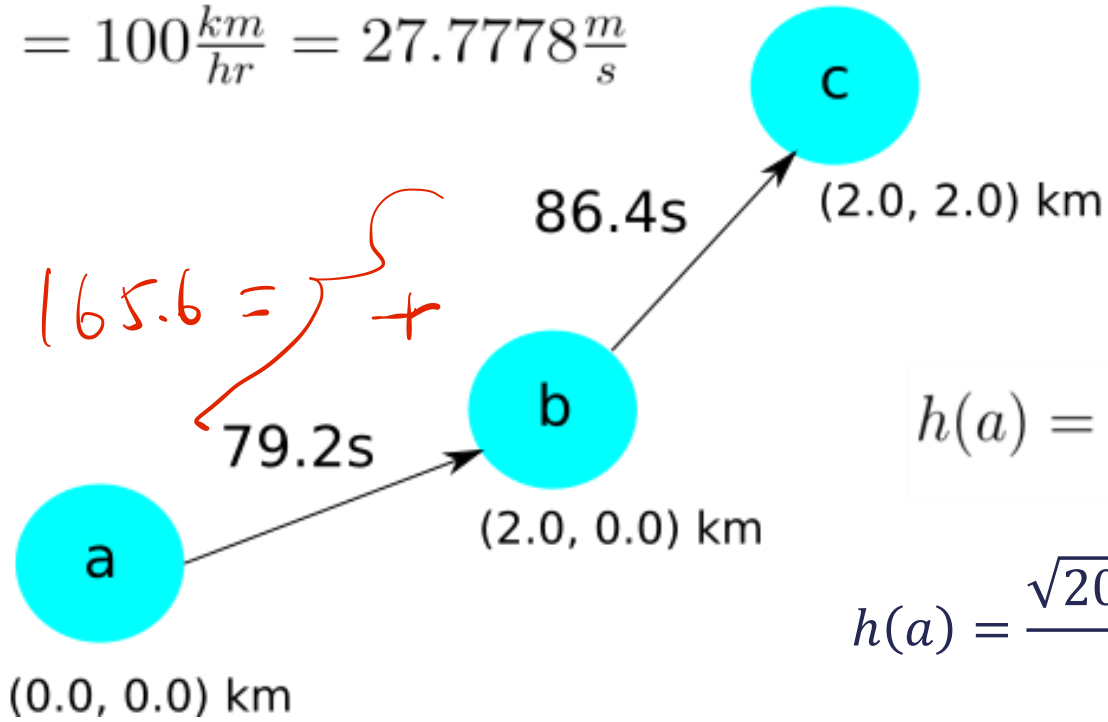


Extensions to Other Factors

- Traffic, speed limits, and weather affect mission planning
- Time rather than distance is better at capturing these factors
- Replace distance edge weights with time estimates

Example

$$v_{max} = 100\frac{km}{hr} = 27.7778\frac{m}{s}$$



$$h(a) = \frac{\sqrt{2^2+2^2}}{v_{max}} = 101.82s$$

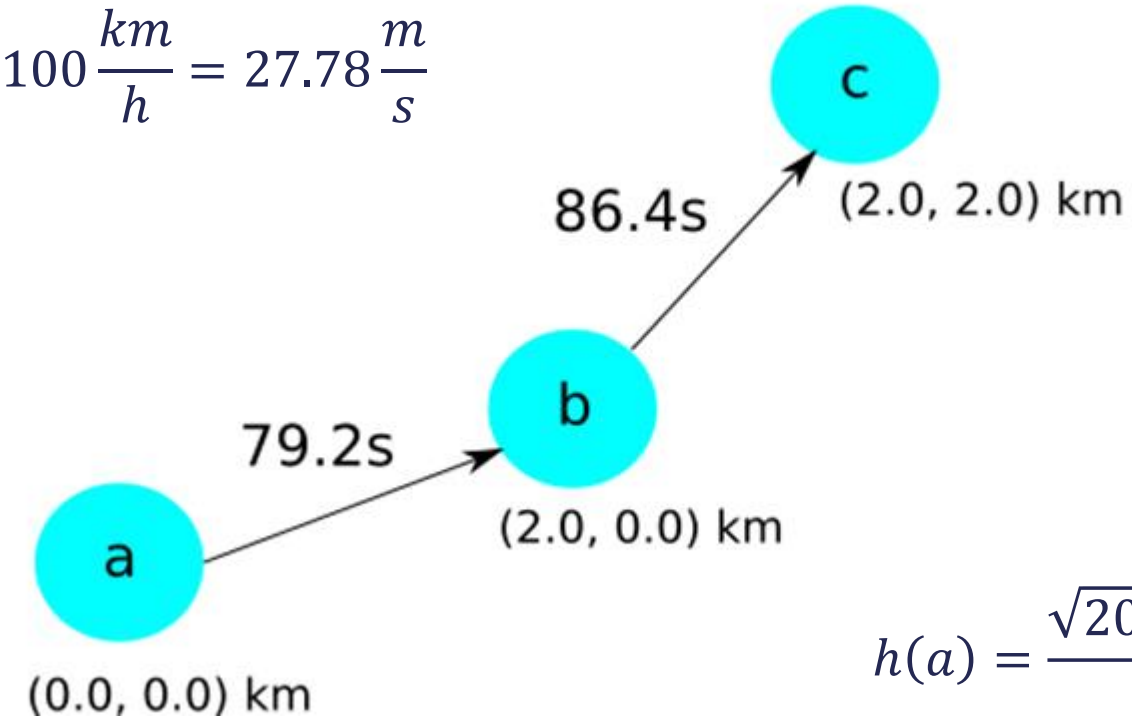
Euclidean distance

$$h(a) = \frac{\sqrt{2000^2 + 2000^2}}{v_{max}} = 101.82s$$

max speed

Example

$$v_{max} = 100\frac{km}{h} = 27.78\frac{m}{s}$$



$$h(a) = \frac{\sqrt{2000^2 + 2000^2}}{v_{max}} = 101.82s$$

Summary

- Introduced Euclidean heuristic, showed it was admissible to our mission planning problem
- Walked through the A* search algorithm
- Discussed how to modify the heuristic to handle travel time rather than distance in our search

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING