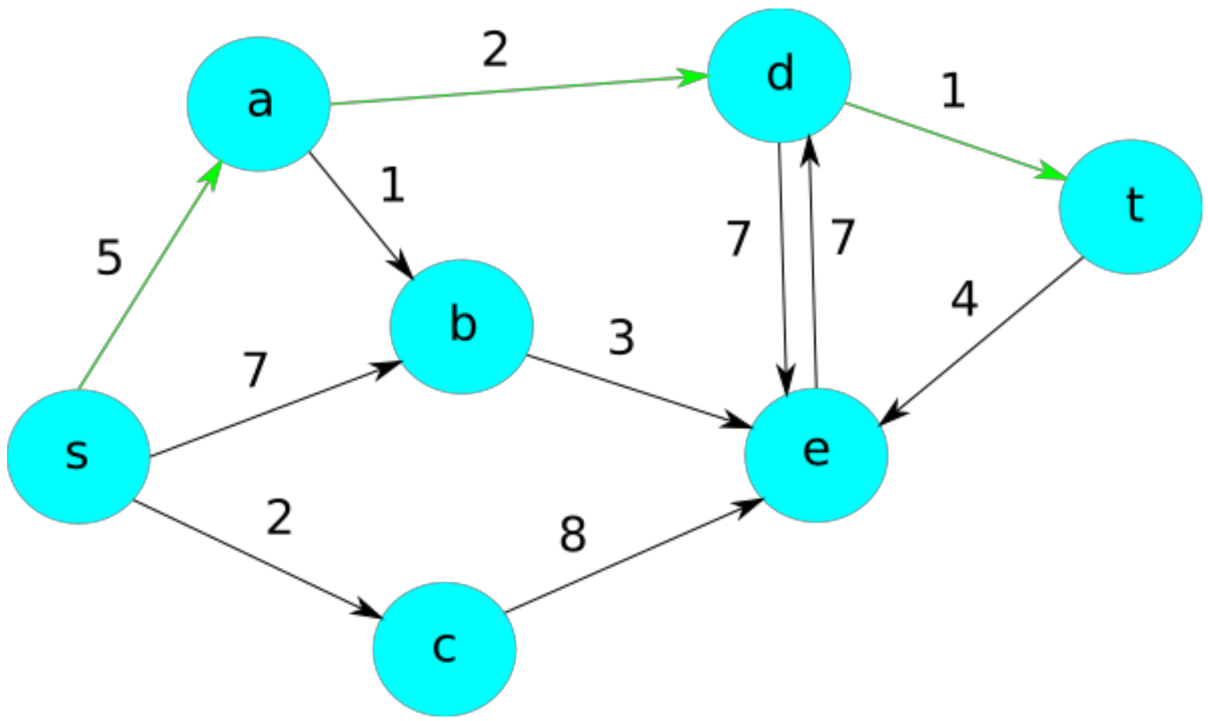
A* Shortest Path Search

Course 4, Module 3, Lesson 3

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

Recall: Dijkstra's for Weighted Graph



Euclidean Heuristic

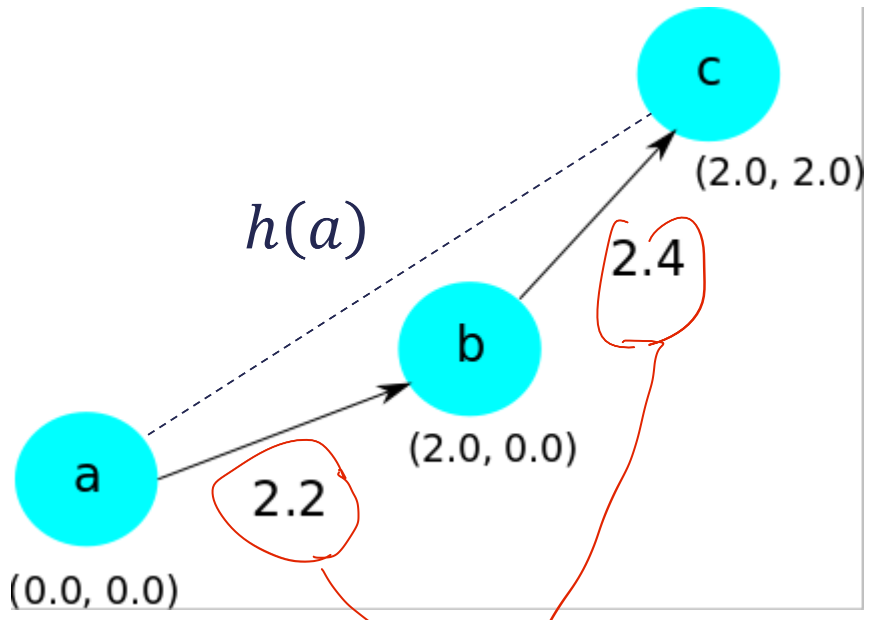
- Exploits structure of the problem
- Fast to calculate
- Straight-line distance between two vertices is a useful estimate of true distance along the graph

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

Euclidean Heuristic - Example

Admissible heuristic
($h \leq$ 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)

```
open ← MinHeap()
closed ← Set()
predecessors ← Dict()
open.push(s, 0)
while !open.isEmpty() do
   u, uCost ← open.pop()
   if isGoal(u) then
      return extractPath(u, predecessors)
   for all v ∈ u.successors()
       if v ∈ closed then
           continue
      uvCost ← edgeCost(G, u, v)
      if v ∈ open then
           if uCost + uvCost + h(v) < open[v] then
                open[v] ← uCost + uvCost + h(v)
                costs[v] ← uCost + uvCost
                predecessors[v] ← u
      else
           open.push(v, uCost + uvCost)
           costs[v] ← uCost + uvCost
           predecessors[v] ← u
   closed.add(u)
```

A* Algorithm

accumulated cost from start

heuristic estimate to the goal

total cost $g = f + h$

if $v \in$ open **then**

  if $u\text{Cost} + uv\text{Cost} + \text{h}(v) < \text{open}[v]$ **then**

    $\text{open}[v] \leftarrow u\text{Cost} + uv\text{Cost} + \text{h}(v)$

    $\text{costs}[\text{v}] \leftarrow u\text{Cost} + uv\text{Cost}$

    $\text{predecessors}[v] \leftarrow u$

else

  $\text{open.push}(v, u\text{Cost} + uv\text{Cost})$

  $\text{costs}[\text{v}] \leftarrow u\text{Cost} + uv\text{Cost}$

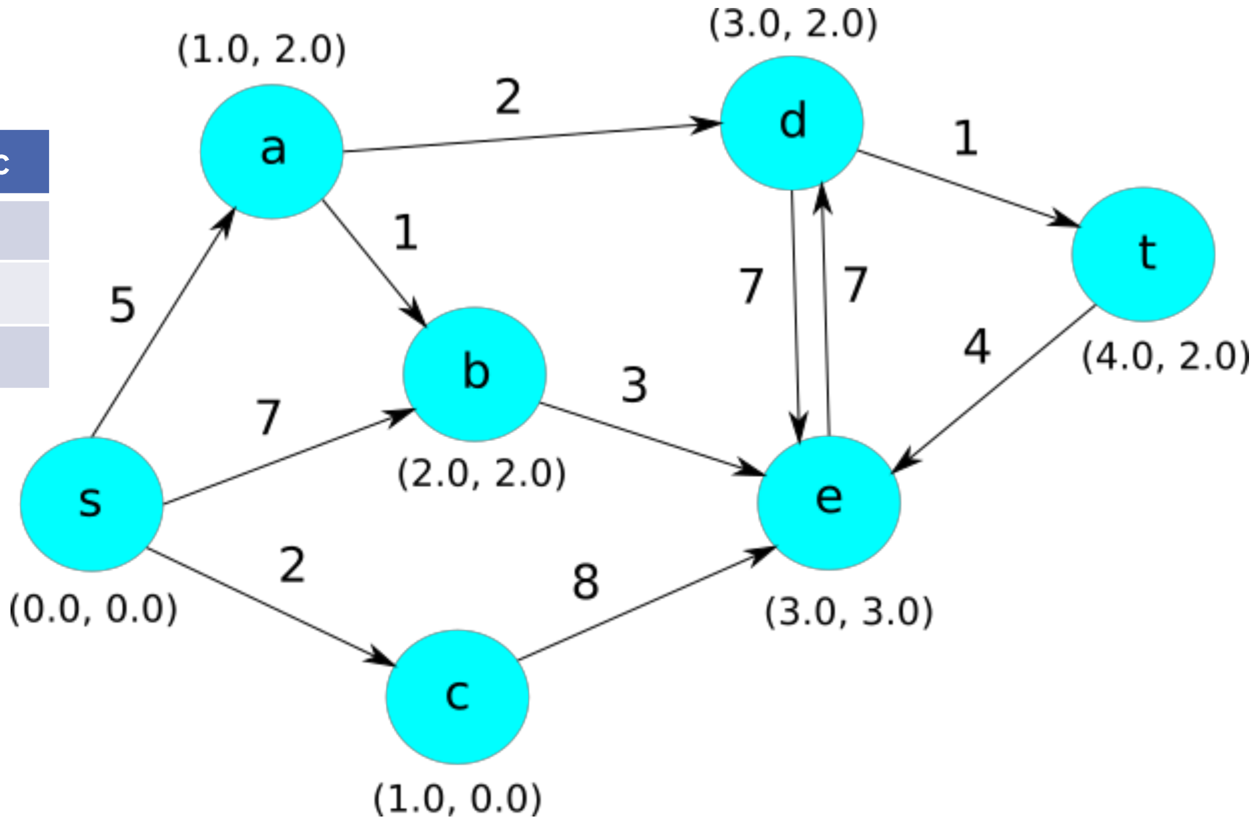
  $\text{predecessors}[v] \leftarrow u$

Example - Origin Node

Open Min Heap:

Node	Cost + Heuristic
s	4.472

Closed Set:

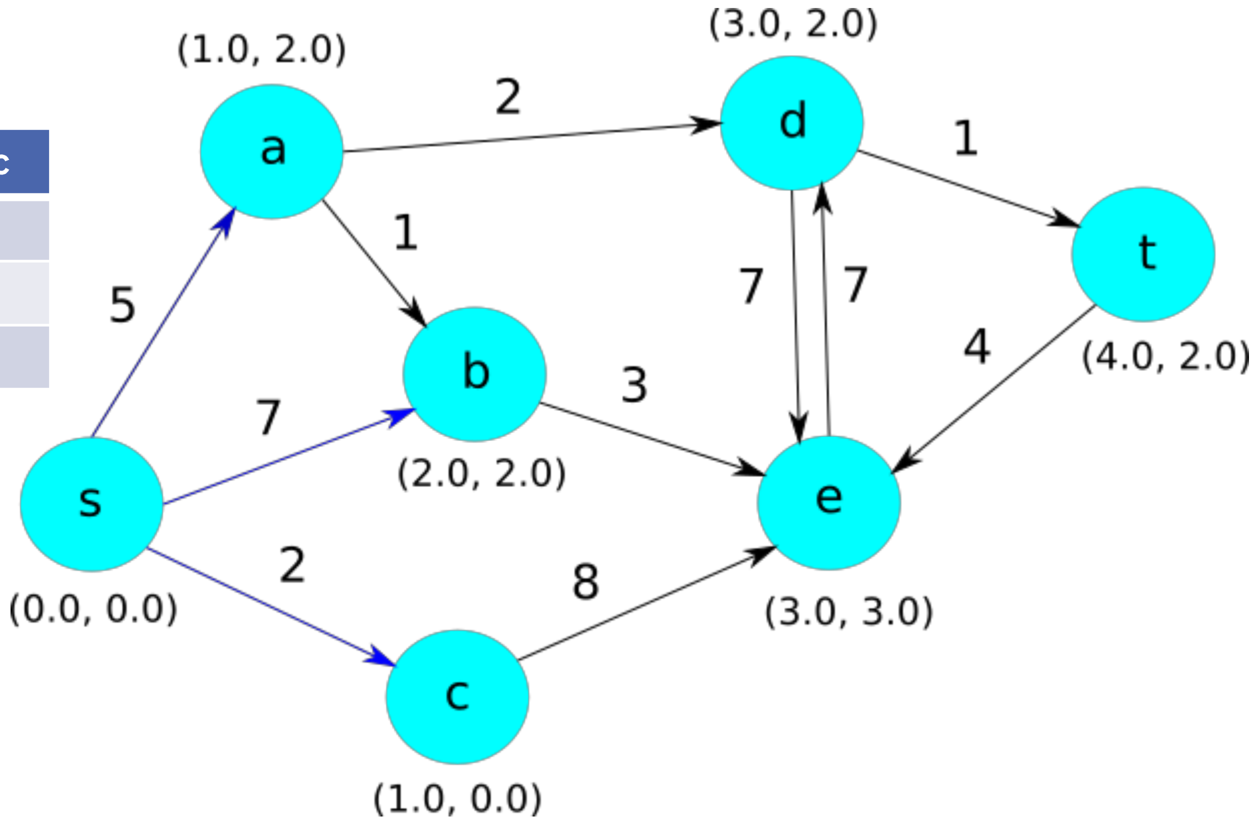


Example - Processing s

Open Min Heap:

Node	Cost + Heuristic
c	5.606
a	8
b	9

Closed Set: s



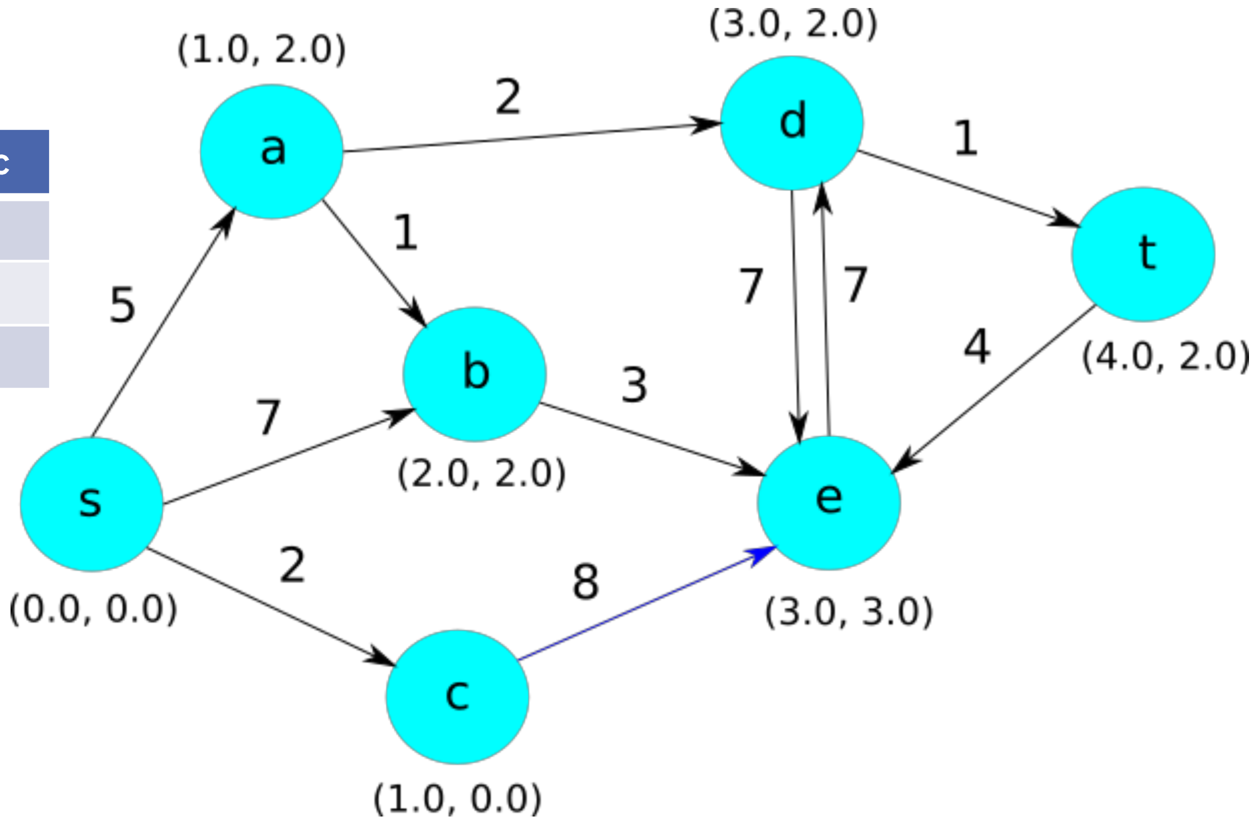
Example - Processing c

Open Min Heap:

Node	Cost + Heuristic
a	8
b	9
e	11.414

Closed Set: s

c

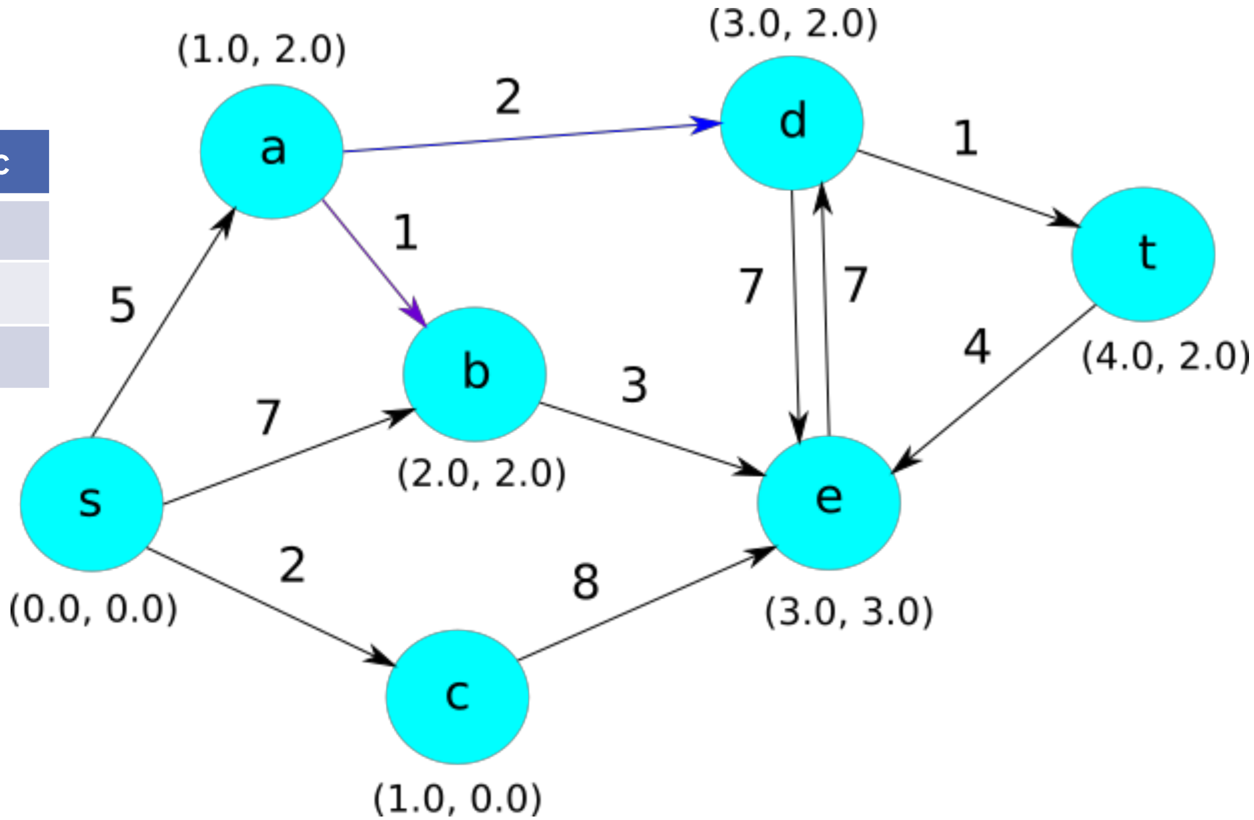


Example - Processing a

Open Min Heap:

Node	Cost + Heuristic
d	7
b	8
e	11.414

Closed Set: s
c
a

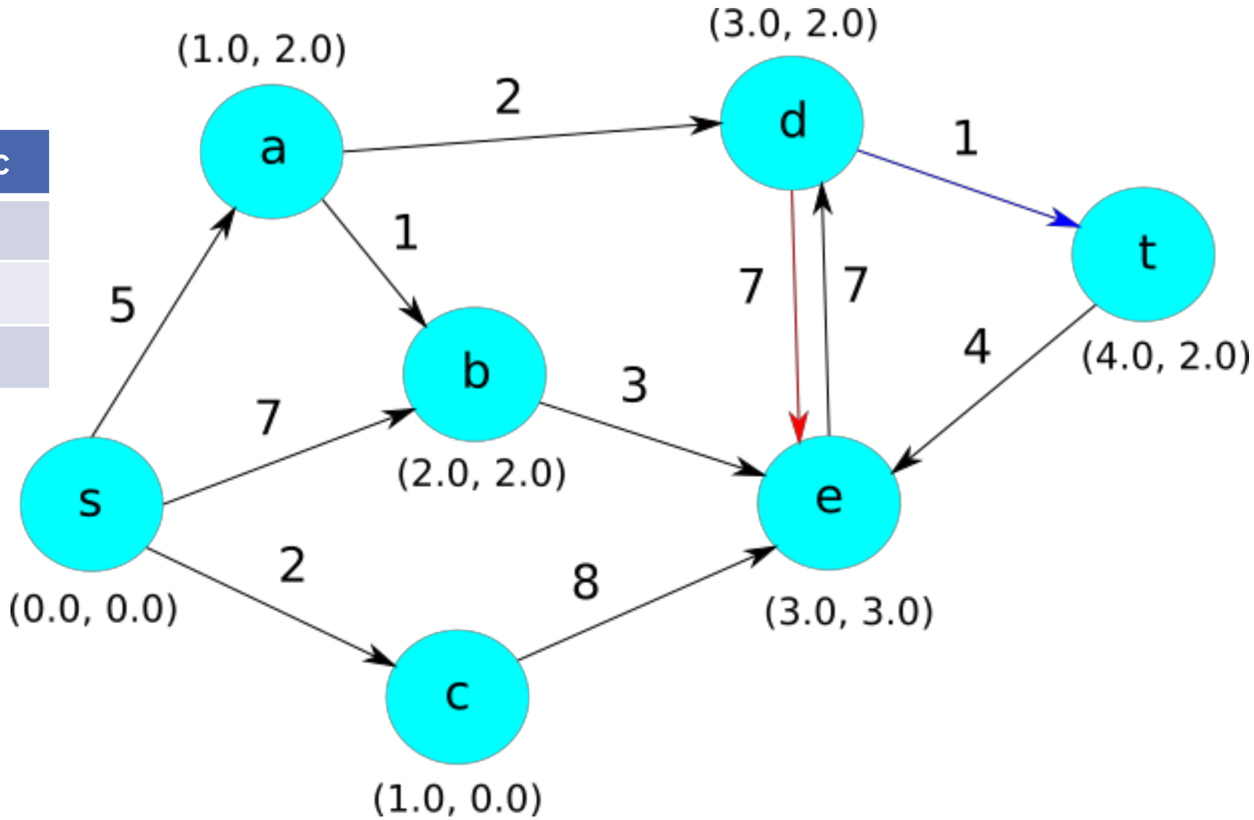


Example - Processing d

Open Min Heap:

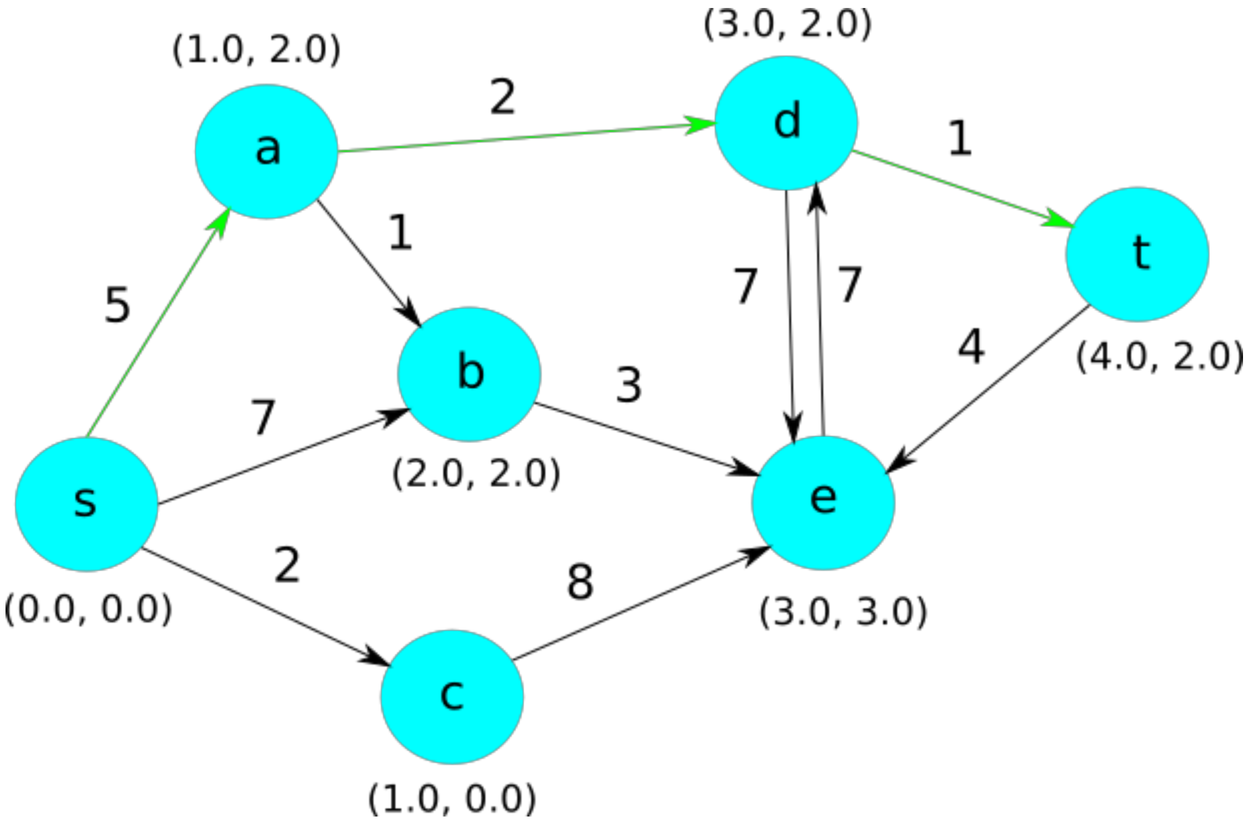
Node	Cost + Heuristic
t	7
b	8
e	11.414

Closed Set: s
c
a
d



Example - Final Path

Final Path: s
a
d
t

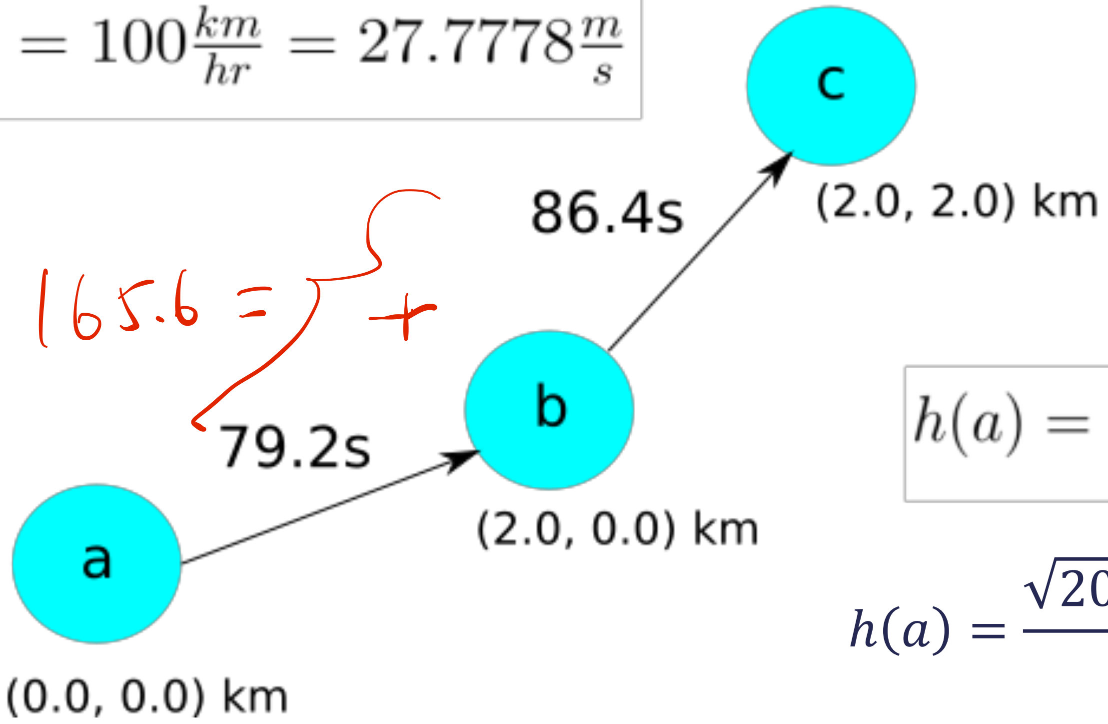


Extensions to Other Factors

- 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

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



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

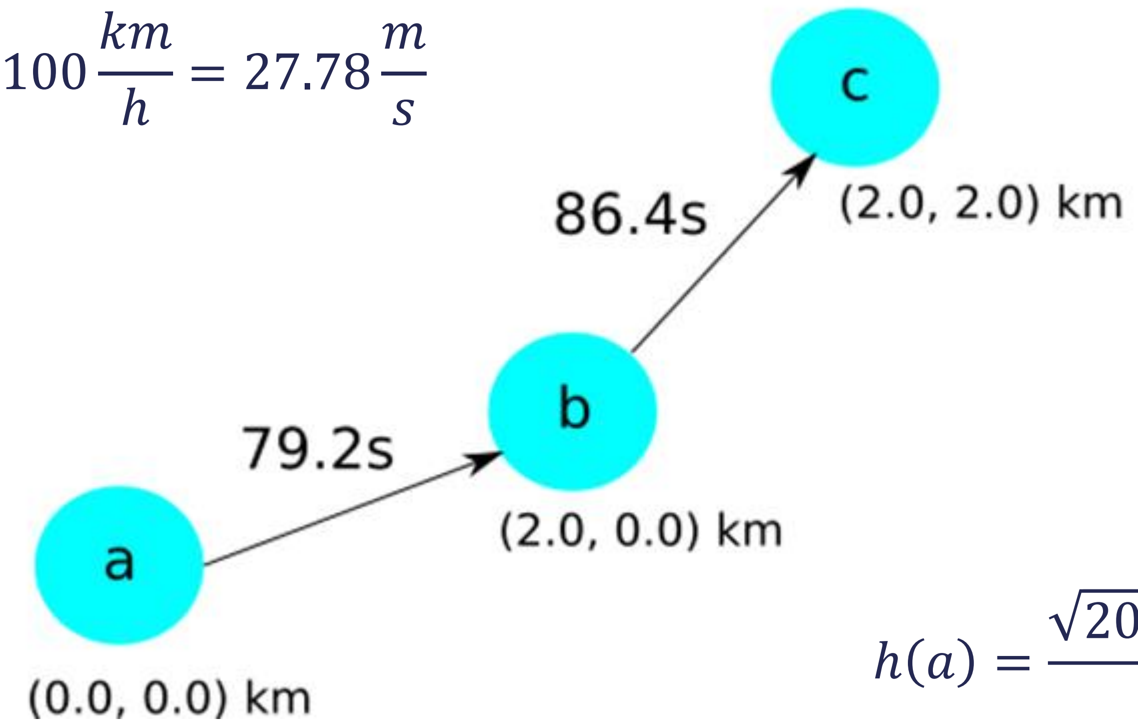
Euclidean distance

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

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