

Graph Search



A* Algorithm



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)
by Christine Alvarado, Mia Minnes, and Leo Porter, 2015.

By the end of this video you will be able to...

- Describe a limitation of Dijkstra's Algorithm
- Apply A* algorithm to a weighted graph
- Write the code for A*

Driving directions from San Diego to Seattle?

San Diego

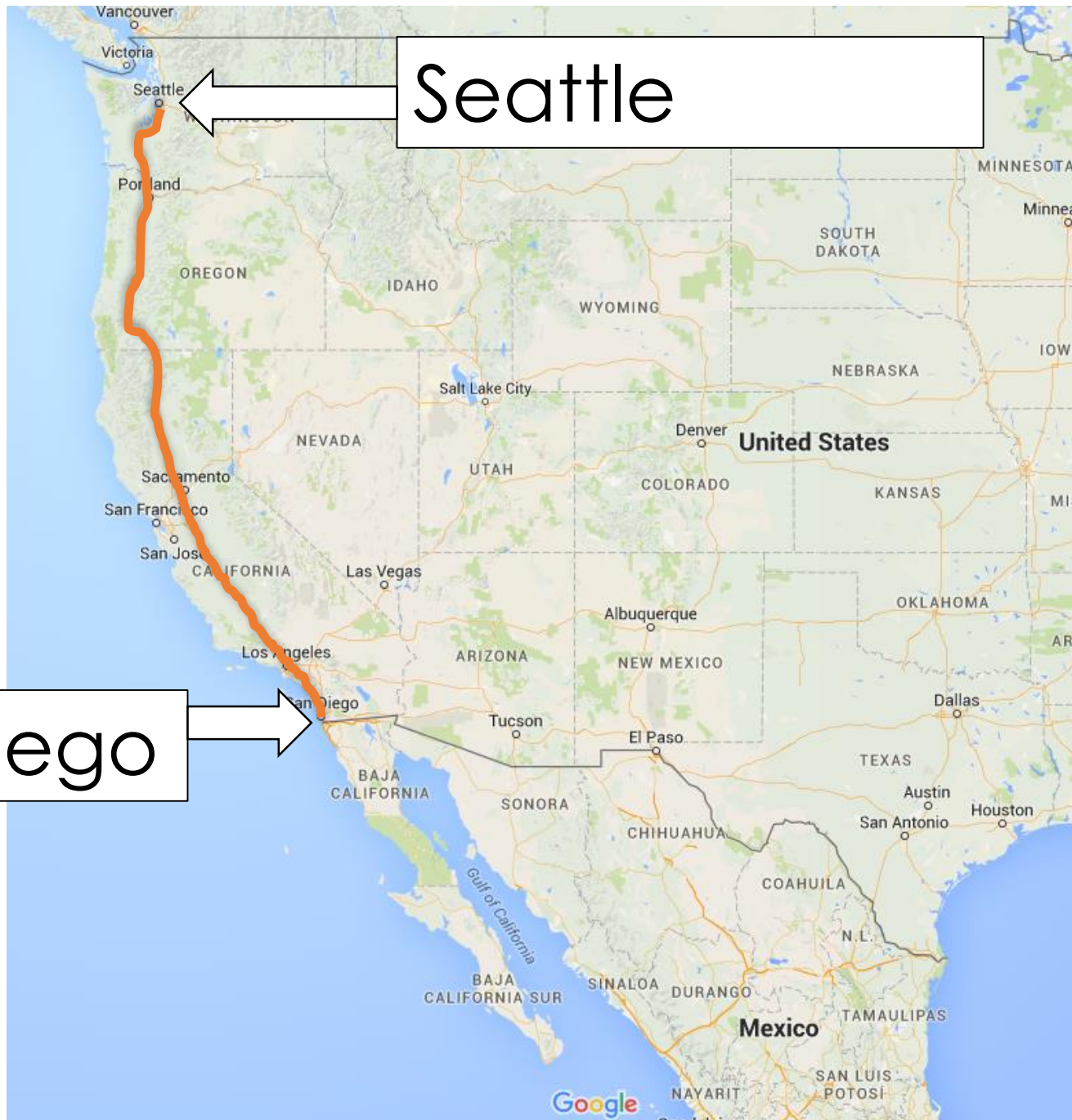
Seattle



Driving directions from San Diego to Seattle?

San Diego

Seattle

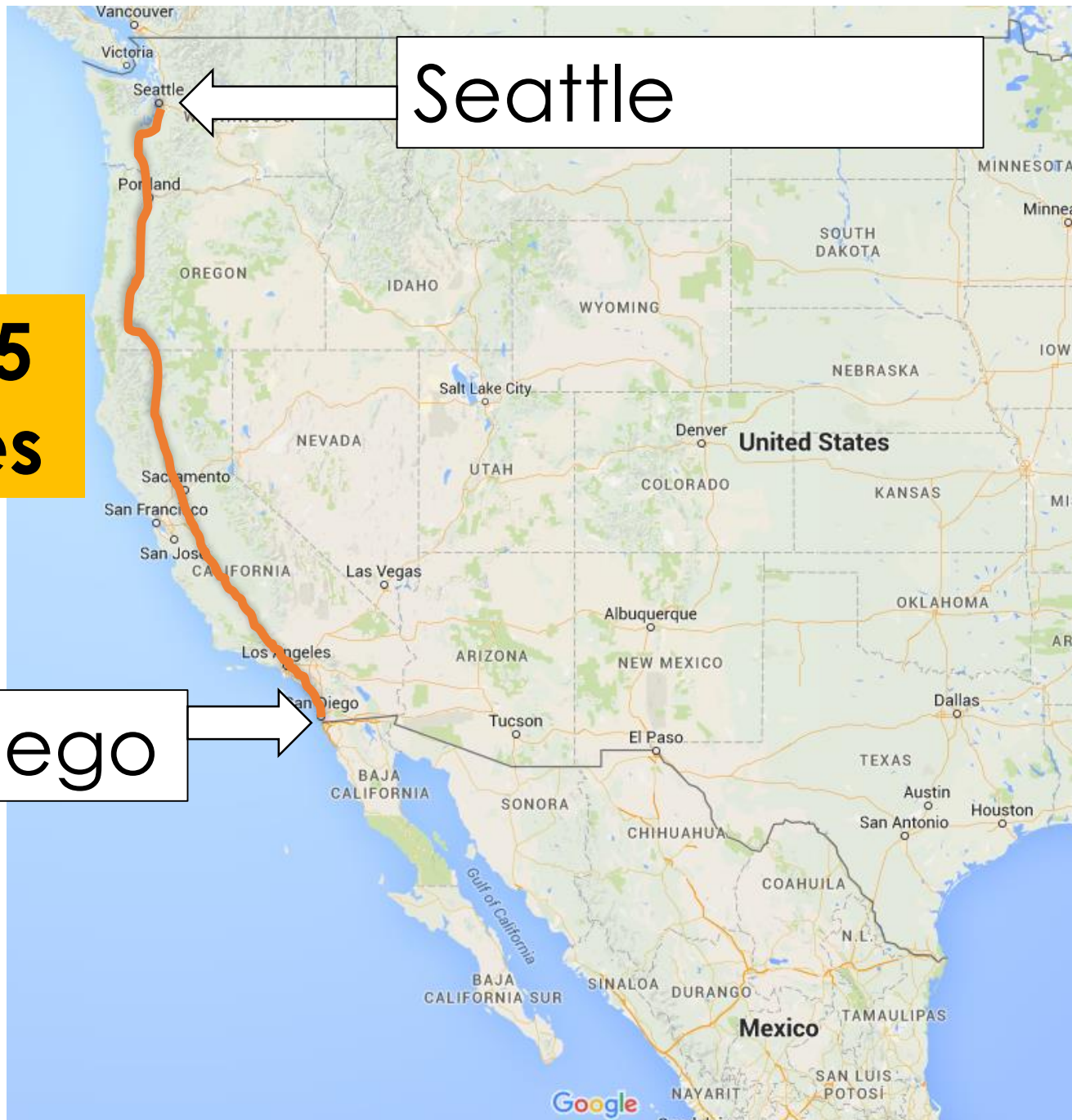


Driving directions from San Diego to Seattle?

**1255
miles**

San Diego

Seattle

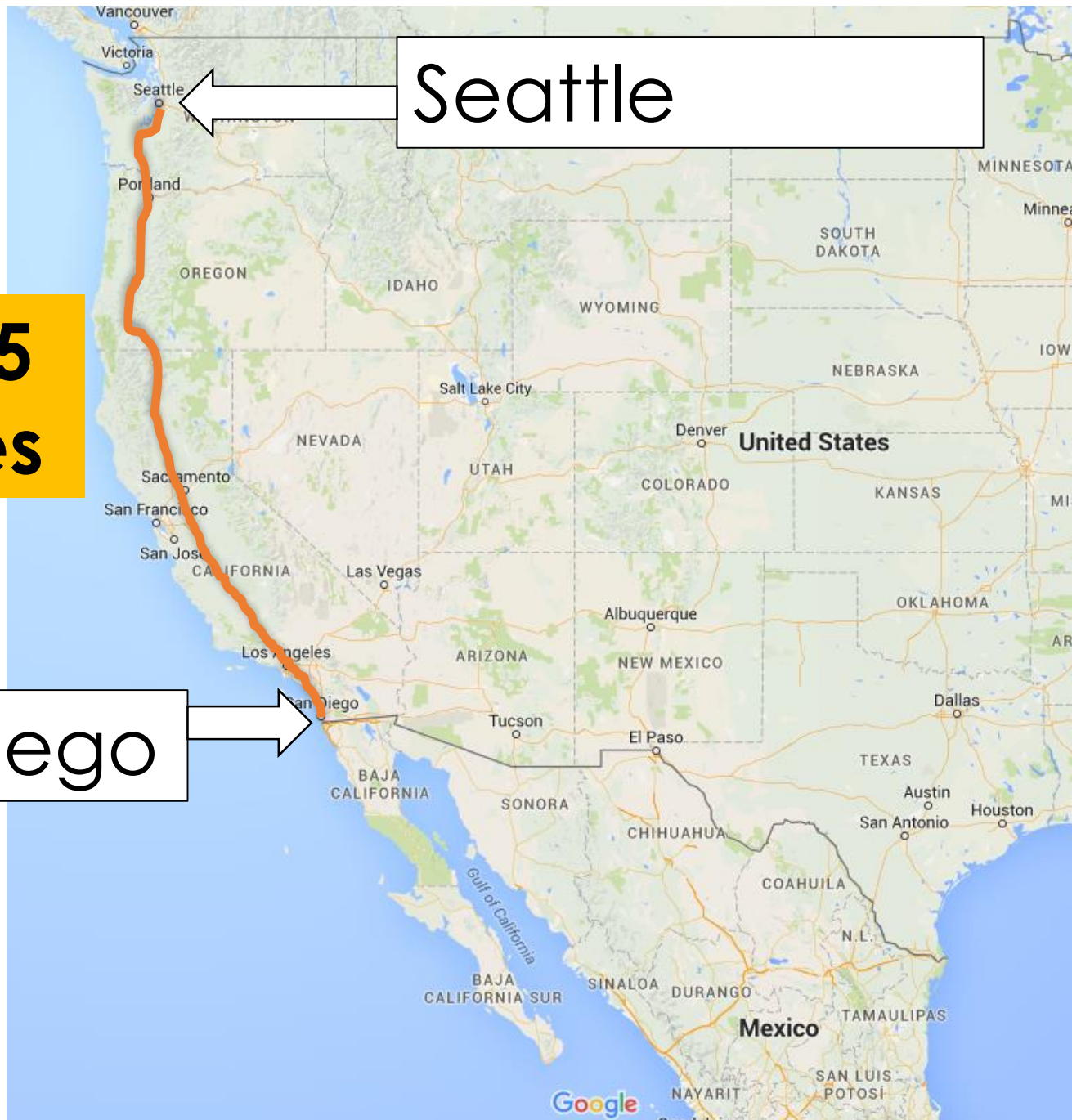


**Dijkstra will find
the shortest
route. But how?**

**1255
miles**

San Diego

Seattle



**Dijkstra will find
the shortest
route. But how?**

Seattle

Denver

Dallas

San Diego

Mazatlán,
Mexico



Seattle

**Why would YOU
have never
considered Dallas?**

Dallas

San Diego

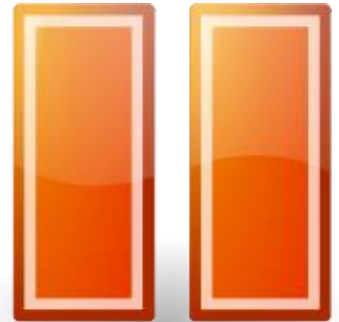


Seattle

**Why would YOU
have never
considered Dallas?**

Dallas

San Diego



Seattle

**Why would YOU
have never
considered Dallas?**

Dallas

San Diego



Seattle

**Going East is the
wrong direction!**

Dallas

San Diego



Seattle

1064
miles

1679
miles

Going East takes us
farther away from
Seattle than we
started!

Dallas

San Diego

Seattle

**We should
consider
distance from
target too!**

**Dijkstra only
considers
distance from
source**

San Diego

Dallas



Dijkstra's Algorithm

- Priority Queue ordering is based on:

$g(n)$: the distance (cost) from start vertex to vertex n

A* Algorithm

- Priority Queue ordering is based on:

$g(n)$: the distance (cost) from start vertex to vertex n

AND

$h(n)$: the **heuristic estimated cost** from vertex n to goal vertex

A* Algorithm

- Priority Queue ordering is based on:

$g(n)$: the distance (cost) from start vertex to vertex n

AND

$h(n)$: the **heuristic estimated cost** from vertex n to goal vertex

$$f(n) = g(n) + h(n)$$

A* Algorithm

- Priority Queue ordering is based on:

$g(n)$: the distance (cost) from start vertex to vertex n

AND

$h(n)$: the **heuristic estimated cost** from vertex n to goal vertex

$$f(n) = g(n) + h(n)$$

Dijkstra can be seen as a special case where $h(n)=0$

A* Algorithm

- Priority Queue ordering is based on:

$g(n)$: the distance (cost) from start vertex to vertex n

AND

$h(n)$: the **heuristic estimated cost** from vertex n to goal vertex

$$f(n) = g(n) + h(n)$$

**Guaranteed to
find shortest
path IF estimate
is never an
overestimate**

Seattle

1064
miles

1679
miles

**Underestimate:
use the exact
distance.**

Dallas

San Diego

A* Algorithm

- Priority Queue ordering is based on:

$g(n)$ the distance (cost) from start vertex to vertex n

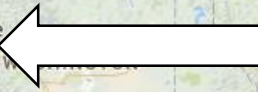
AND

$h(n)$ the **heuristic estimated cost** from vertex n to goal vertex

$$f(n) = g(n) + h(n)$$



Seattle



San Diego





Seattle

Sacramento

Las Vegas

San Diego





Seattle

Sacramento

$$f(n) = 504 + 625 = 1129$$

Las Vegas

$$f(n) = 331 + 871 = 1202$$

San Diego

A* Algorithm

- Priority Queue ordering is based on:

$g(n)$ the distance (cost) from start vertex to vertex n

AND

$h(n)$ the heuristic estimated cost from vertex n to goal vertex

$$f(n) = g(n) + h(n)$$

**Just change the
priority function!**