OPTIONAL TOPIC
(not on exam or assignments)
Finding shortest routes
on a map.

Dijkstra’s algorithm
A*

US road network
24 M nodes
58 M edges
5 μs - 63 μs
query response time

O(m + n log m)
number of nodes
number of edges

> 1 sec

how to reduce runtime
- reduce m
- parallelize the search
- store some results
\[ d(s, t) \]
\[
\min \sum d(s, u) + d(u, v) + d(v, t)
\]
\[
\text{s's access node} \quad \text{t's access node}
\]

All distances for all pairs of transit nodes are precomputed and stored.

Local Queries: Dijkstra's, A*

next problem
- distance from \( s \) to \( u \)
- what will be \( s \)'s transit nodes?
- nodes with high degree or candidates
- space out evenly
To identify the transit nodes:
Get the set of nodes \( V_{inner} \)

such that
there exists a shortest path from some node in \( V_{C} \) to some node in \( V_{outer} \) which passes through \( V \).

tradeoff: time vs memory