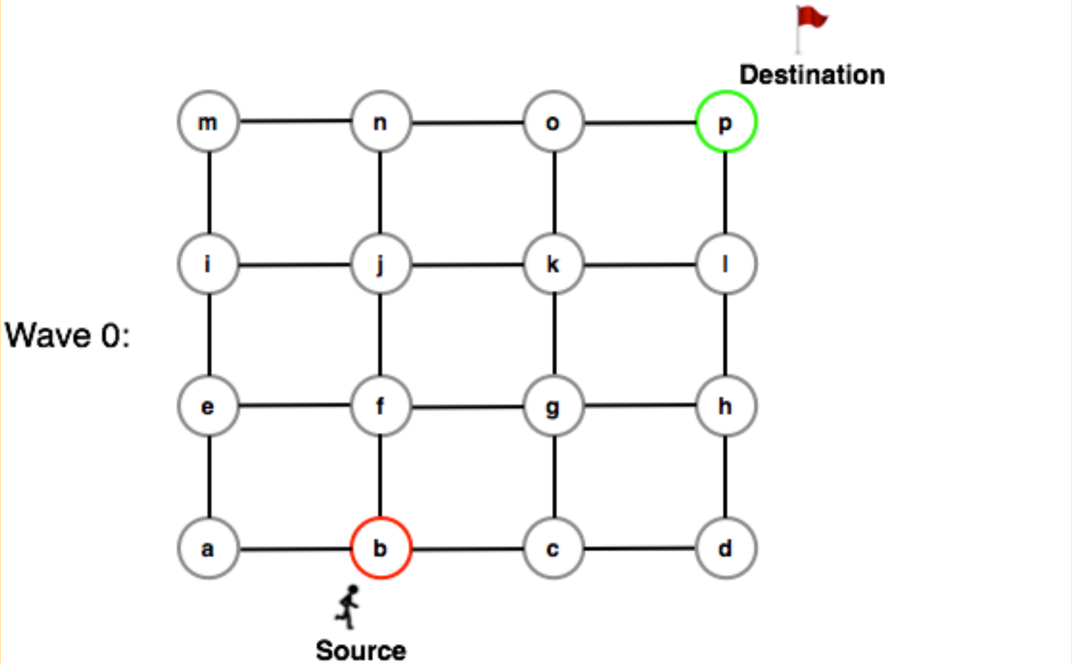
Learning Objectives

- After this segment, students will be able to
  - List 2 algorithms for shortest path queries
  - Compare those two algorithms

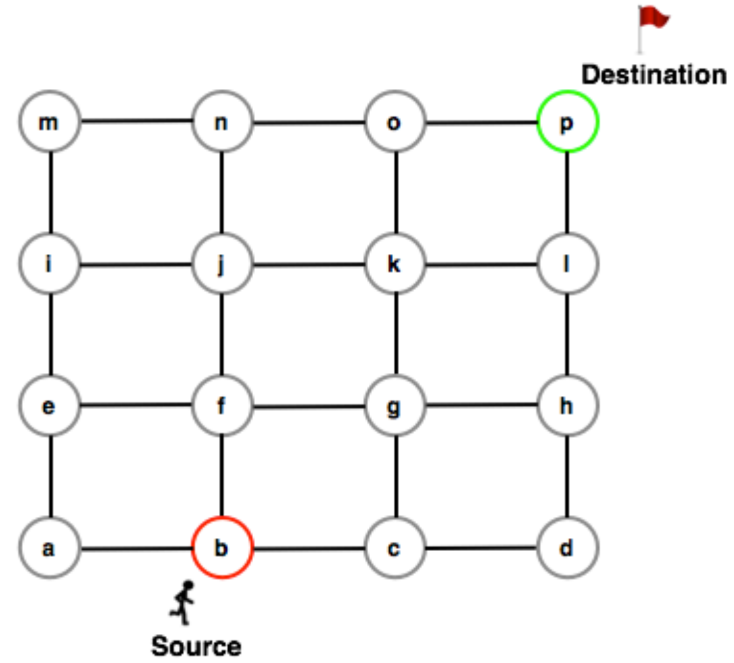
Shortest Path Algorithms

- Iterate
  - Expand most promising descent node
    - Dijkstra’s: try closest descendent to self
    - A* : try closest descendent to both destination and self
  - Update current best path to each node, if a better path is found
- Till destination node is expanded

Dijkstra’s vs. A*

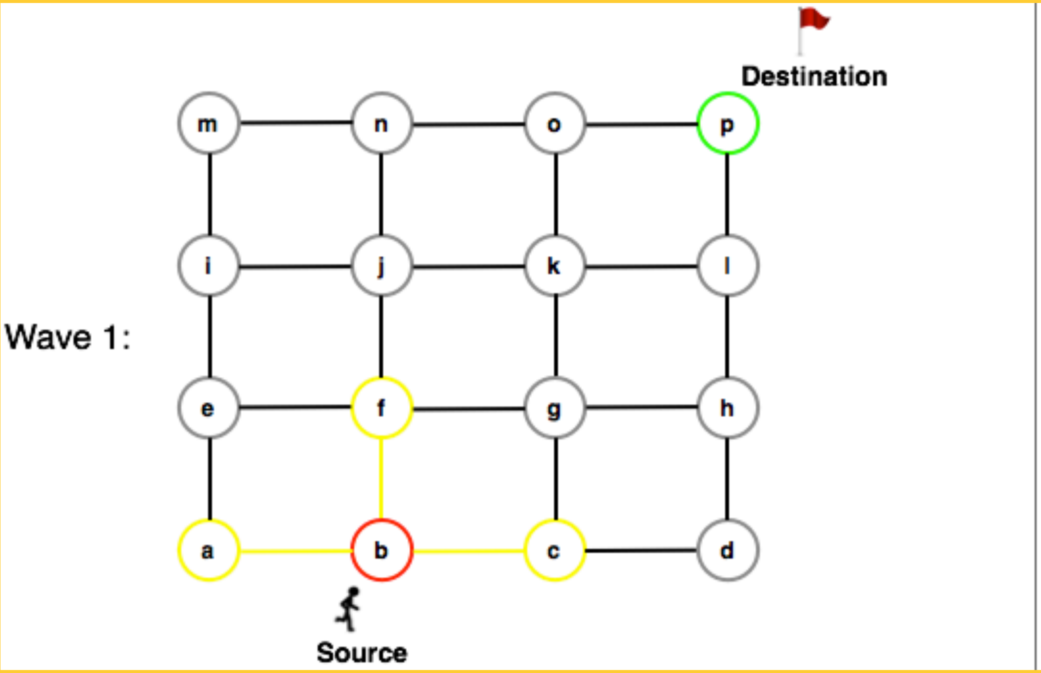


Dijkstra’s Algorithm

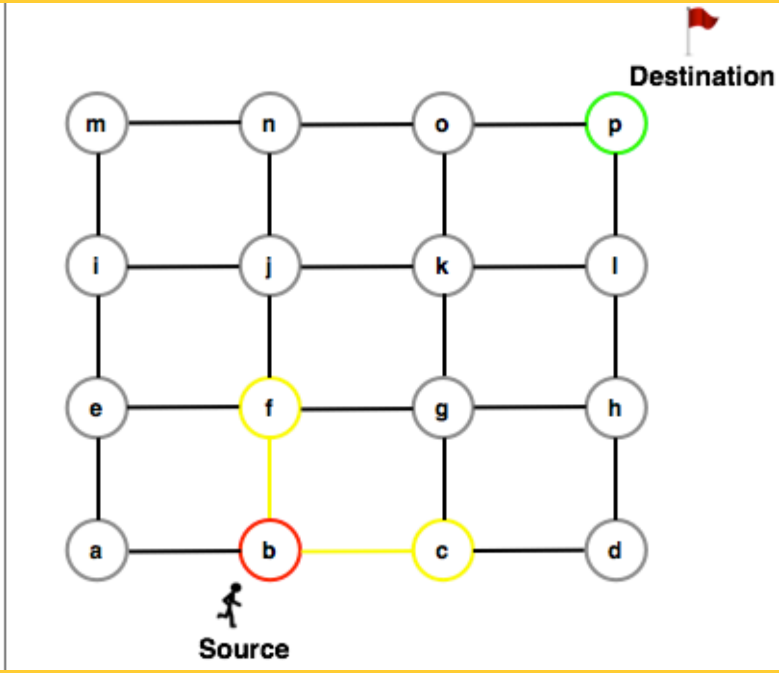


A* Algorithm

Dijkstra’s vs. A*



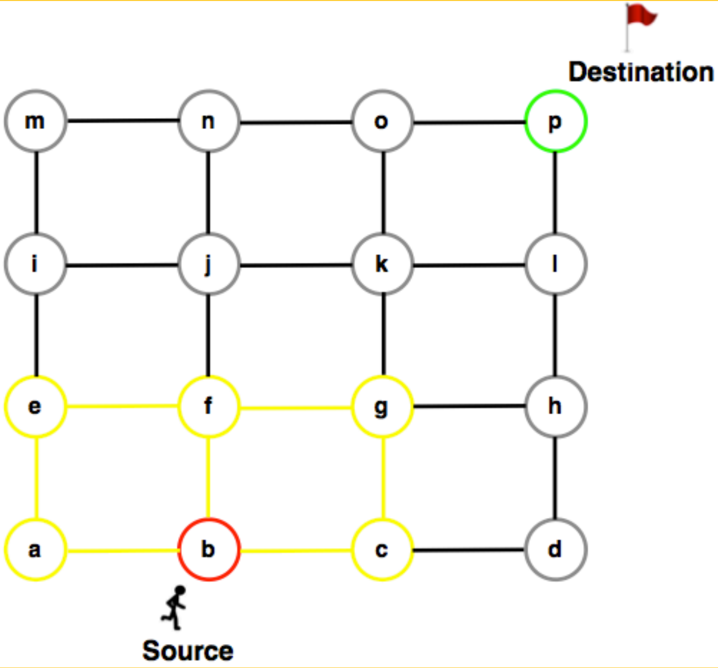
Dijkstra’s Algorithm



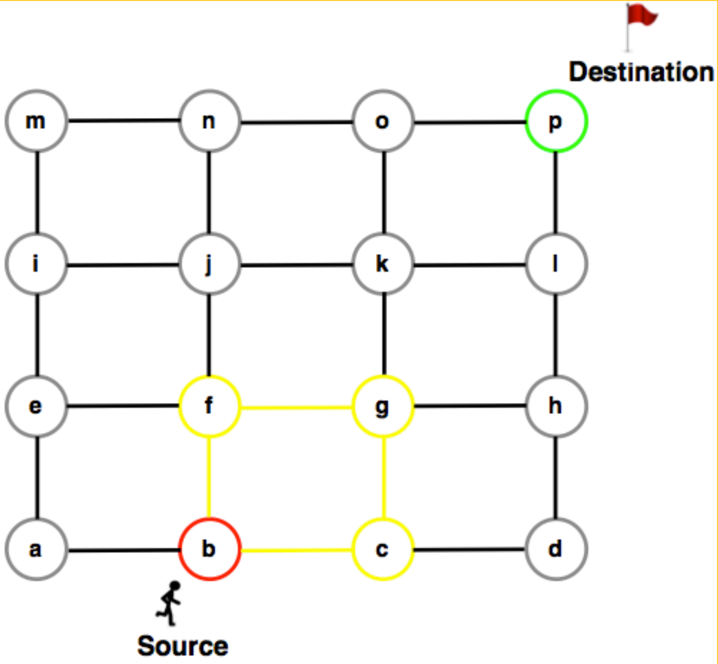
A* Algorithm

Dijkstra's vs. A*

Wave 2:

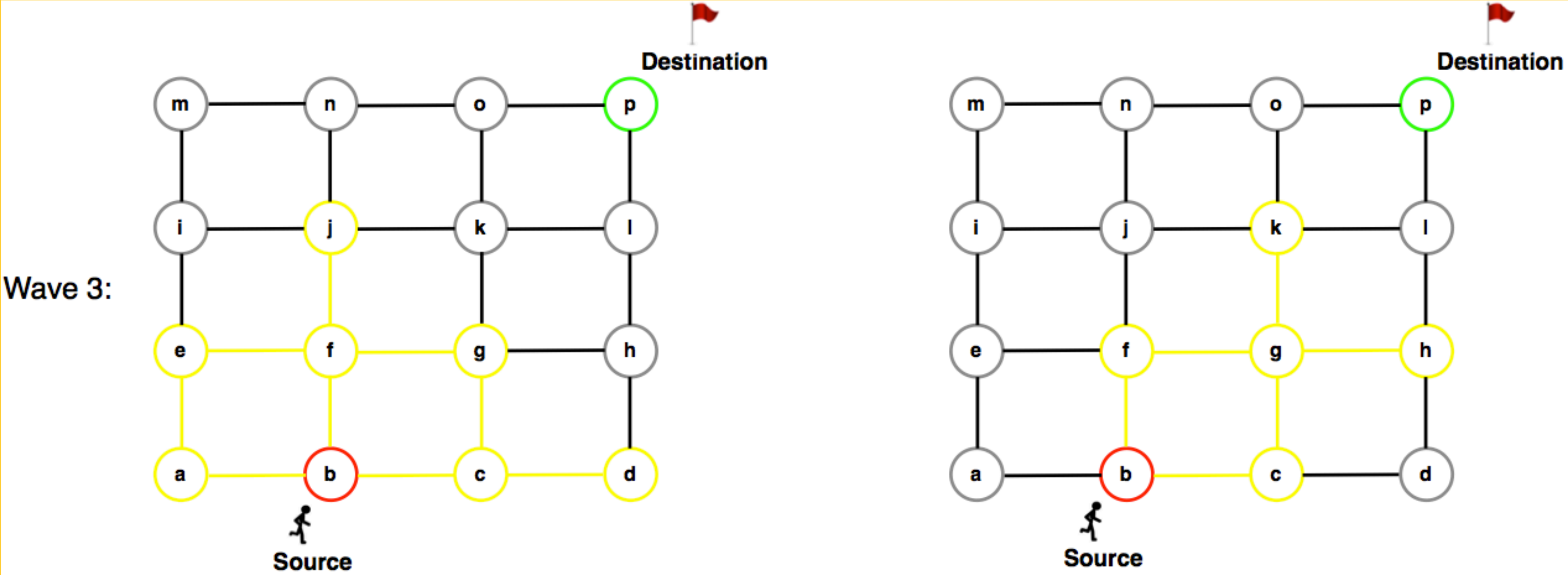


Dijkstra's Algorithm



A* Algorithm

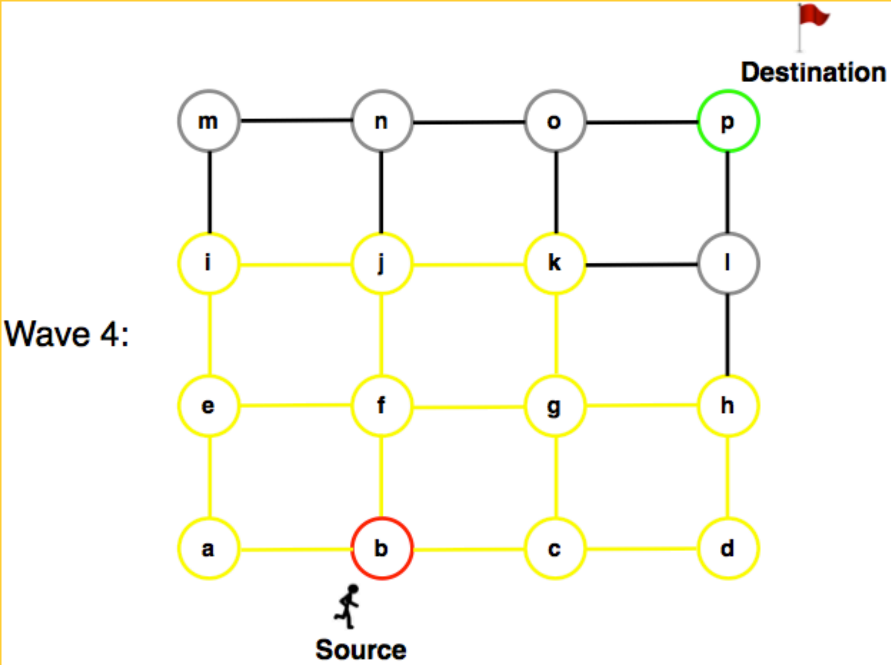
Dijkstra's vs. A*



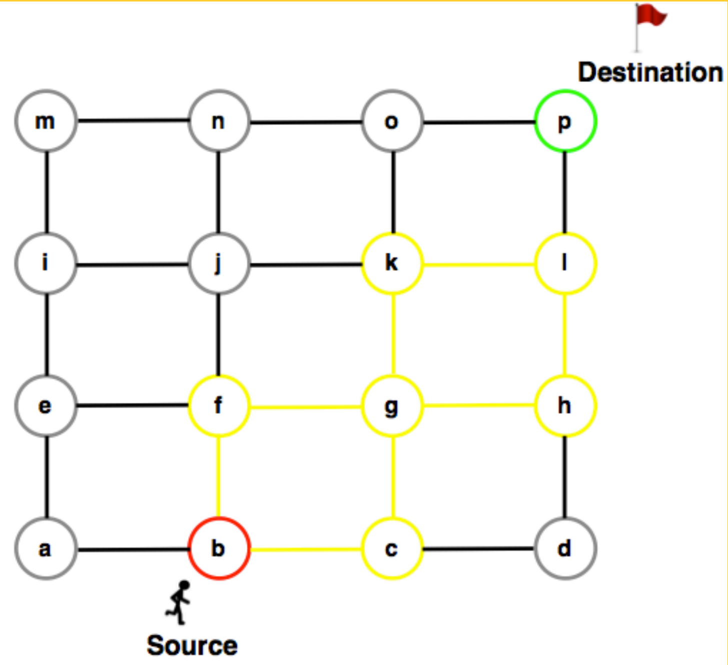
Dijkstra's Algorithm

A* Algorithm

Dijkstra's vs. A*



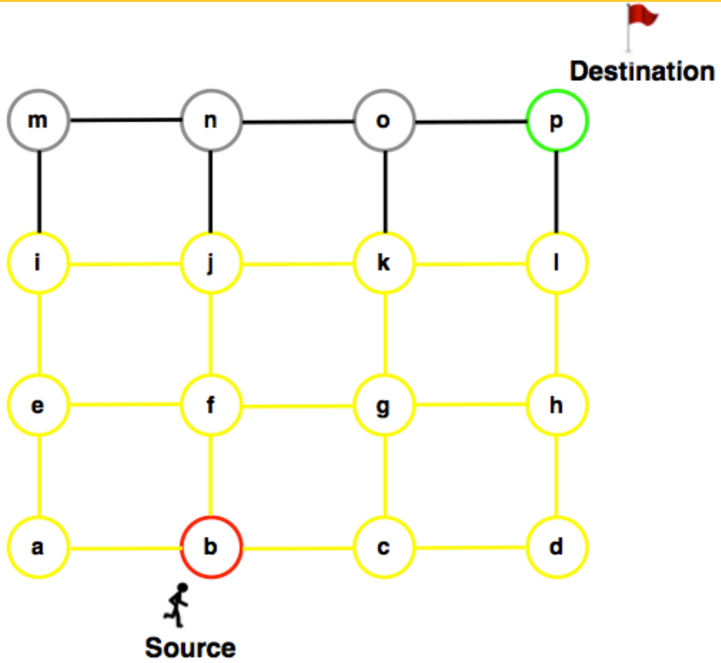
Dijkstra's Algorithm



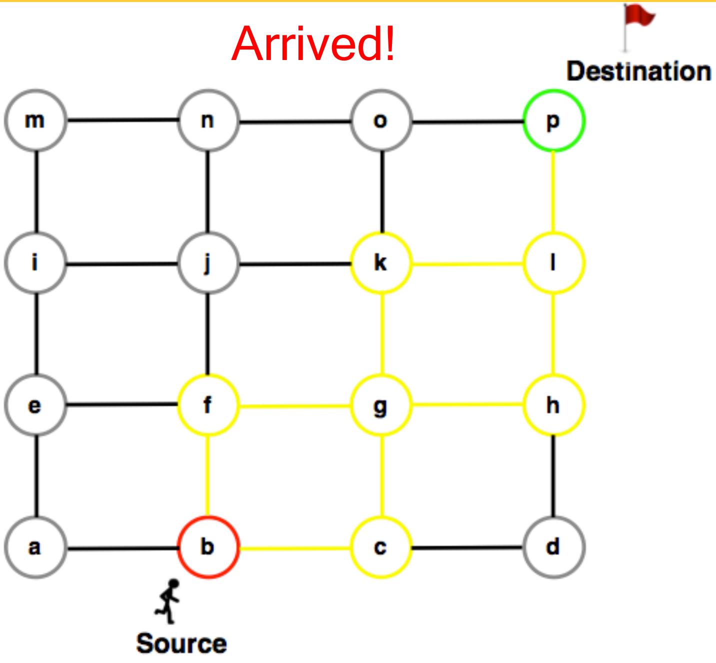
A* Algorithm

Dijkstra's vs. A*

Wave 5:



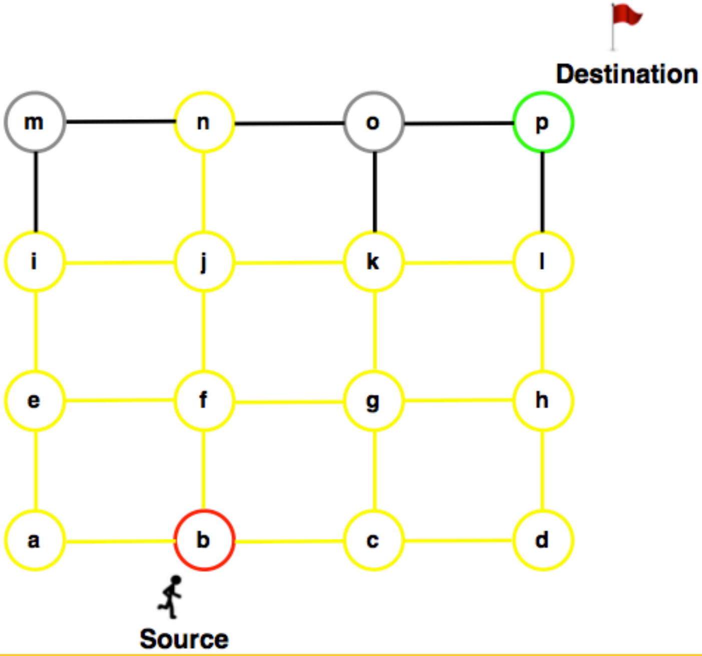
Dijkstra's Algorithm



A* Algorithm

Dijkstra's vs. A*

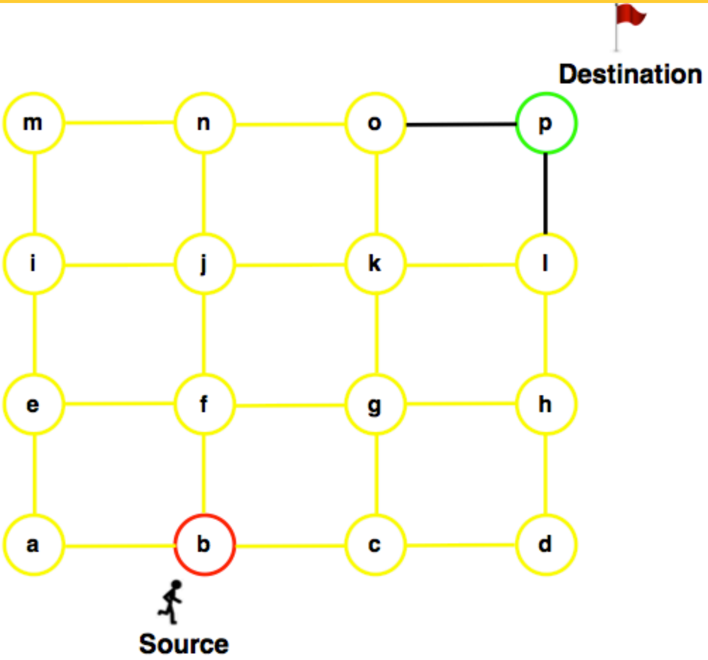
Wave 6:



Dijkstra's Algorithm

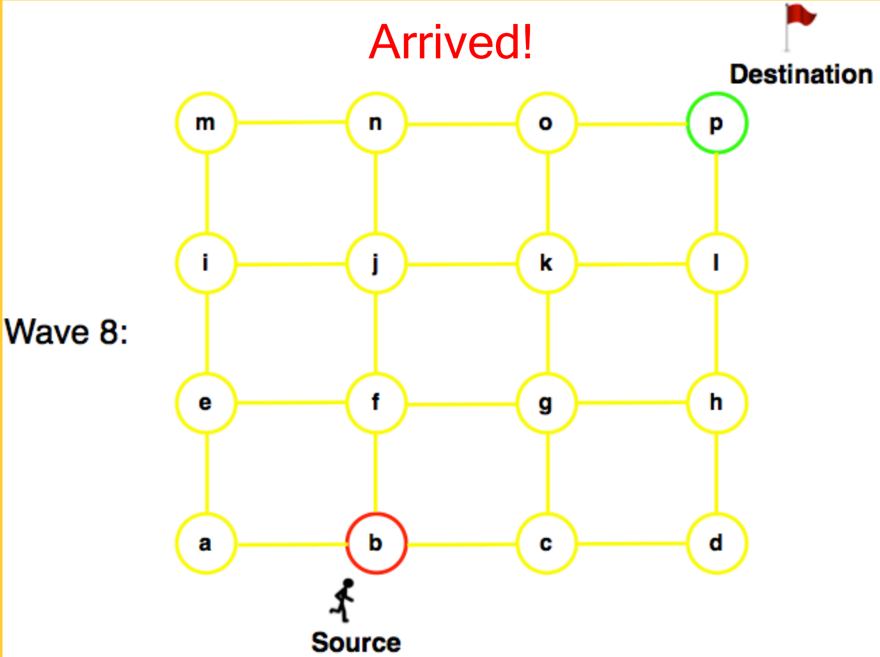
Dijkstra's vs. A*

Wave 7:



Dijkstra's Algorithm

Dijkstra's vs. A*



Dijkstra's Algorithm

Shortest Path Algorithms

- Iterate
  - Expand most promising node
    - Dijkstra’s: try closest descendent to self
    - A* : try closest descendent to both destination and self
  - Update current best path to each node, if a better path is found
- Till destination node is expanded

- Correct assuming
  - Sub-path optimality
  - Fixed, positive and additive edge costs
  - A* : underestimate function