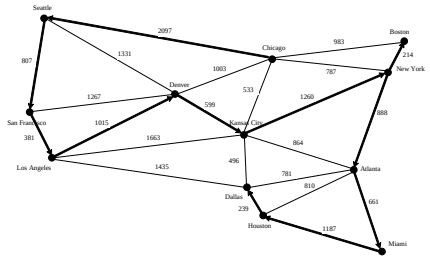
Dijkstra's Algorithm

Extra Material Not On Final Exam

1

Finding the Shortest Path

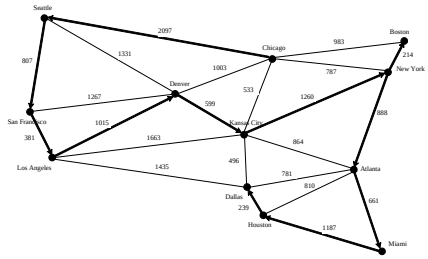


What is the shortest path between two cities?
E.g. Seattle to Dallas…

2

2

… or Finding the Minimal Cost Spanning Tree



This is equivalent to finding the shortest path from a start node (city) to every other city

3

3

Dijkstra's Algorithm

- How? (main ideas)
  - Keep an estimate of the path cost to every node
    - That never underestimates
    - Initial estimate is ∞ for every node except start node
    - Initial estimate is 0 for start node
  - Keep a dynamic priority queue of unvisited nodes
    - So that you can process the unvisited node with the lowest cost
    - Notice this is different from breadth-first or depth-first

4

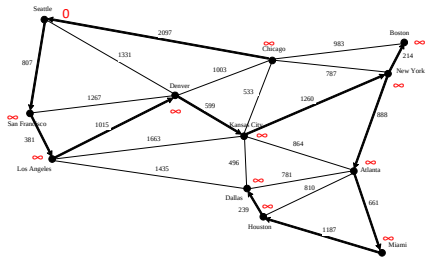
Algorithm

Dijkstra (node s, graph g)

- Set path cost of every node in g to ∞
- Set path cost of s to 0
- Create min priority queue PQ containing all nodes, based on path cost
- While PQ not empty
  - Set current node to pop(PQ)
  - For all nodes n adjacent to current node
    - New Path Cost (n) = Path Cost (current) + Edge Cost (current, n)
    - If New Path Cost (n) < Path Cost (n)
      - Set Path Cost(n) = New Path Cost (n)
      - Make current as parent of n // for spanning tree
      - Update PQ

5

Example: Shortest Paths from Seattle

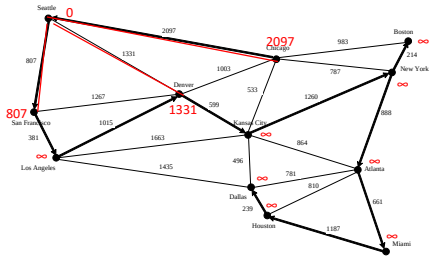


What is the shortest path between two cities?
E.g. Seattle to Dallas…

6

6

Example: Shortest Paths from Seattle

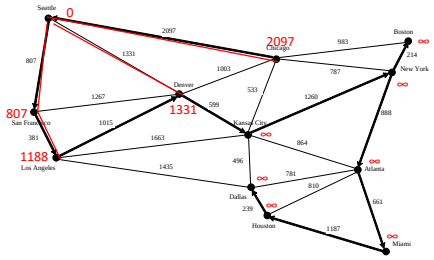


What is the shortest path between two cities?
E.g. Seattle to Dallas…

7

7

Example: Shortest Paths from Seattle

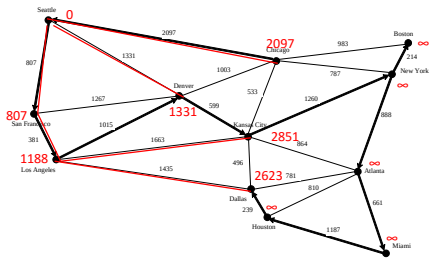


Top of priority queue is now San Francisco (807)
Expand to LA, but Denver path is more expensive than going straight to Denver
Cost to LA is Sea->SF + SF->LA

8

8

Example: Shortest Paths from Seattle

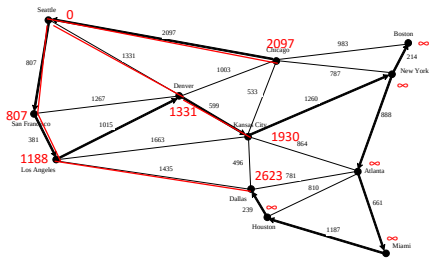


Top of priority queue is Los Angeles (1188)
Expand to KC, Dallas, but Denver path is still more expensive

9

9

Example: Shortest Paths from Seattle

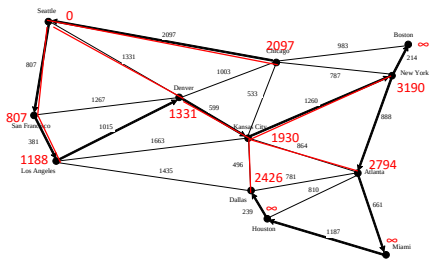


Top of priority queue is Denver (1331)
Produces shorter path to KC, takes over as parent
Chicago, SF, etc paths are too long

10

10

Example: Shortest Paths from Seattle

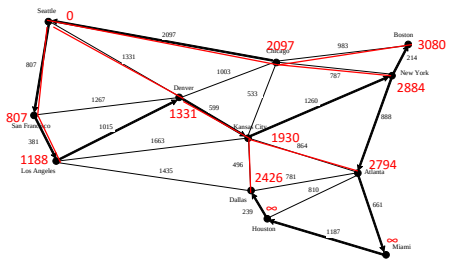


Top of priority queue is Kansas City (1930)
Produces shorter path to Dallas, takes over as parent
New paths to New York & Atlanta

11

11

Example: Shortest Paths from Seattle

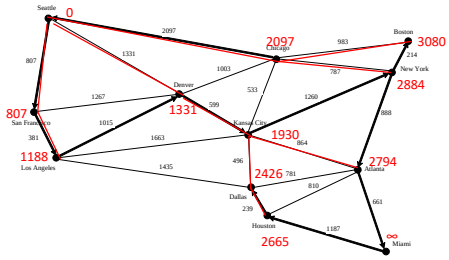


Top of priority queue is Chicago (2097)
Takes over path to New York
New path to Boston

12

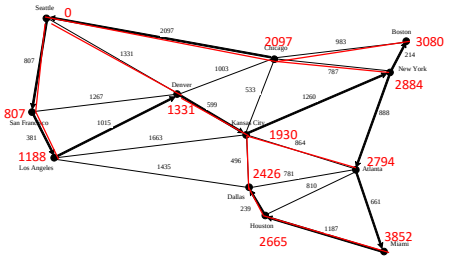
12

Example: Shortest Paths from Seattle



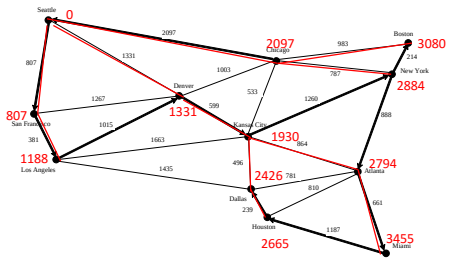
Top of priority queue is Dallas (2426)
New path to Houston

Example: Shortest Paths from Seattle



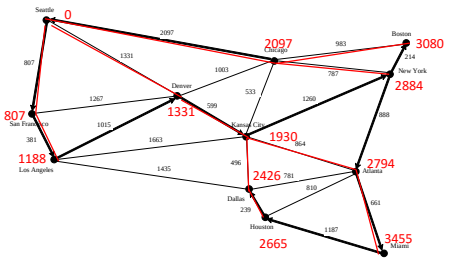
Top of priority queue is Houston (2665)
New path to Miami

Example: Shortest Paths from Seattle



Top of priority queue is Atlanta (2794)
Takes over path to Miami

Example: Shortest Paths from Seattle



Now we process New York (2884), Boston (3080), and Miami (3455), but there are no more changes

Summary

- Produces minimal cost spanning tree
  - Given a starting node
- If goal is minimum cost path from A to B
  - Run with A as start node
  - Finish when B is top of priority queue
- Assumes positive costs
  - What can happen with negative costs?
  - Can it be fixed?