Single source Shortest path

Robin Jadoul

OLYMPIADE BELGE D'INFORMATIQUE
BELGISCHE INFORMATICA-OLYMPIADE

March 13, 2016

Table of Contents

BFS

Dijkstra

Bellman-Ford

BFS

Breath First Search

- Unweighted graphs
- Works with a queue (*First in - First out*)
- Take the next unexplored node from the frontier
- Goal reached: shortest path found
- Refer to *Unit 14 - Graph traversals*
- $O(|V| + |E|)$

Table of Contents

BFS

Dijkstra

Bellman-Ford

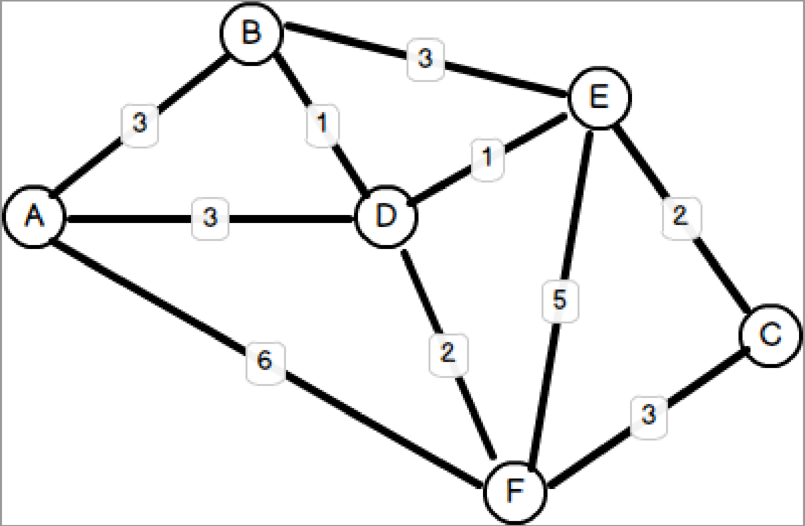
Dijkstra

Single source Shortest path

- Weighted graphs
- Doesn't work with negative edges / cycles
- Works with a priority queue (*min-heap*)
- Only add to the heap if shorter than current shortest = a *relax* operation
- Don't process again (cf. BFS)

Dijkstra

Example



Dijkstra

Extra remarks

- Degrades to BFS on unweighted graphs
- all distances from start needed → don't stop until everything is visited
- This is the *greedy* approach
- $O(|E| \times \log |V|)$

- Why do negative weights fail?

Table of Contents

BFS

Dijkstra

Bellman-Ford

Bellman-Ford

Single source Shortest path

- Weighted graphs - with negative weights
- Can detect negative weight cycles
- No special datastructure needed
- Keep the current minimal distance from the start to each node
- $|V| - 1$ times:
- Relax every edge
  $dist[v] = min(dist[v], dist[u] + weight[u, v])$
- After $|V| - 1$ iterations: every minimal distance from source found
- $O(|V||E|)$

Bellman-Ford

Negative weight cycle detection

- Run the normal Bellman-Ford algorithm
- Do one more iteration
- If the iteration finds a shorter path $\Rightarrow$ a negative weight cycle was found