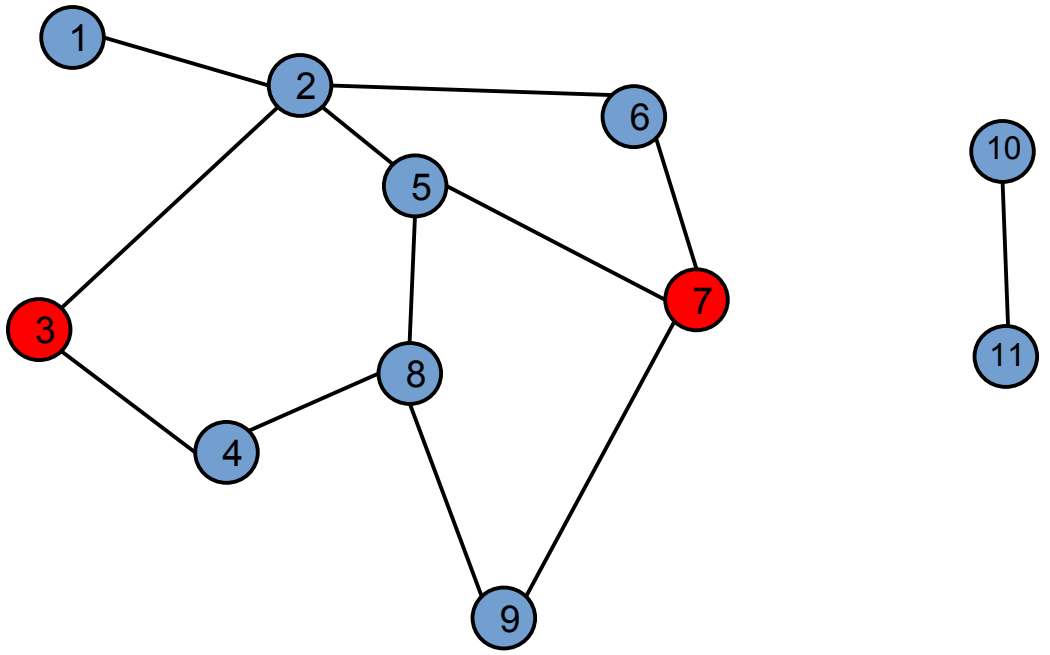
Minimal paths in graphs

Simeon Monov

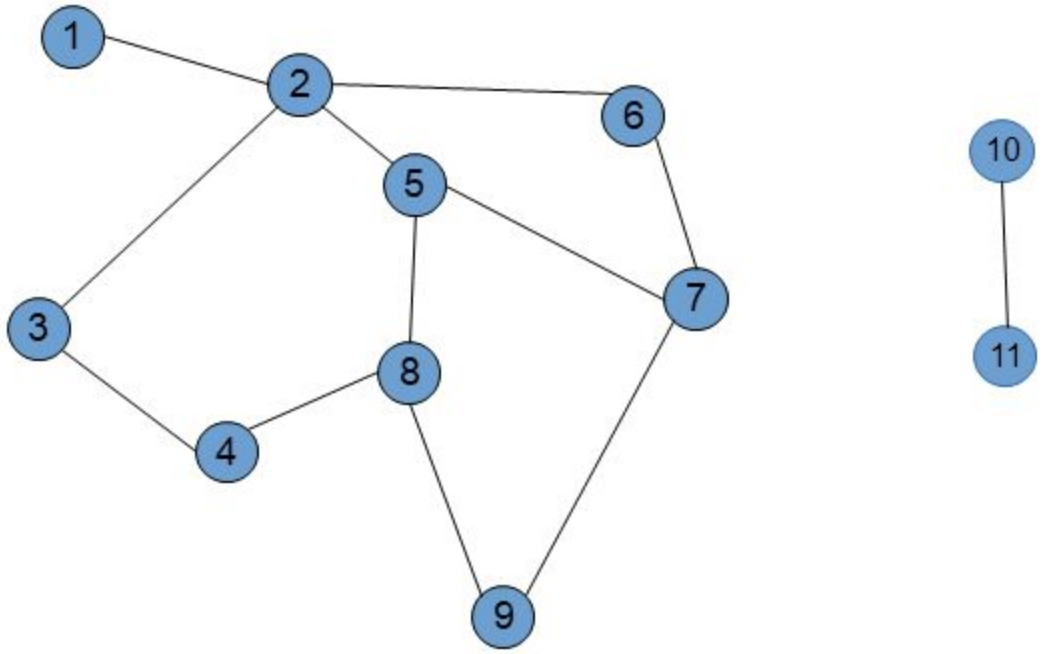
Minimal path in unweighted graph

- Find minimal path between two vertices **a** and **b** by number of vertices/edges
- Using Depth-First-Search (DFS)
- Using Breadth-First-Search (BFS)
- The graph can be both directed or undirected



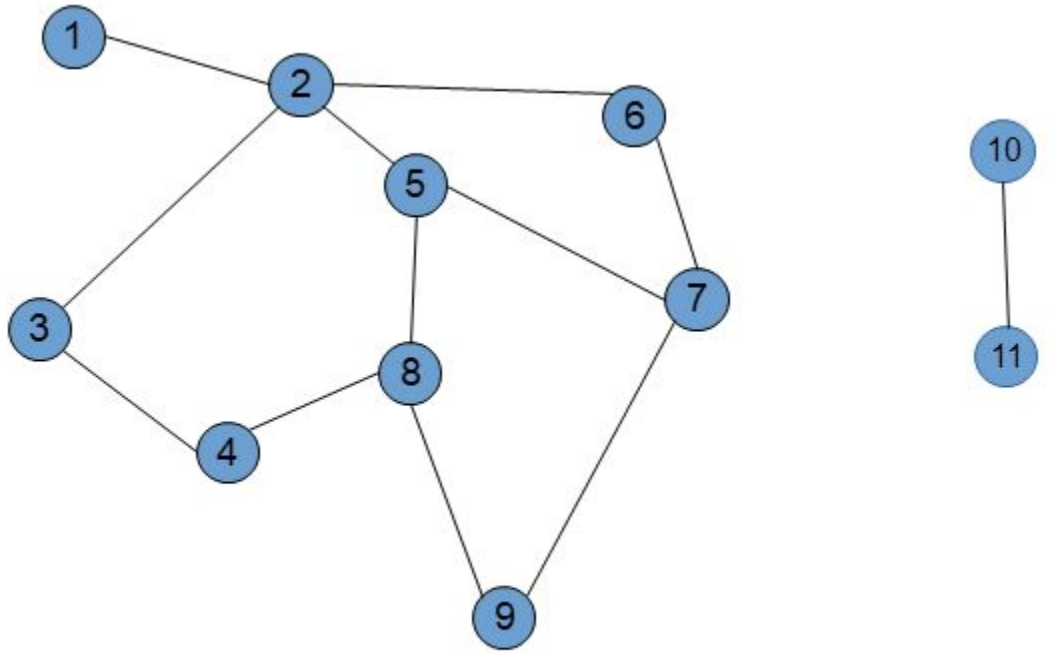
Minimal path unweighted – DFS

- Run DFS from **a**.
- On each step add to the current path.
- If current vertex is already in the path, return back.
- When **b** is reached check if current path is better than before and save it. Return back.
- If **b** is never reached (all vertices are traversed from **a**) there is no solution.



Minimal path unweighted – BFS

- Run **BFS** from **a** until reach **b**
- Save predecessor vertices during the execution in array or hashtable
- If **b** is never reached there is no solution



Minimal path in weighted graph

- Find minimal path between two vertices **a** and **b** by sum of all weights
- Using Depth-First-Search (DFS)
- ~~Using Breadth-First-Search (BFS)~~ – not possible. Why?
- Using Dijkstra's algorithm (BFS can be see as modification of Dijkstra's algorithm for unweighted graphs)