**Combinatorial Optimization**

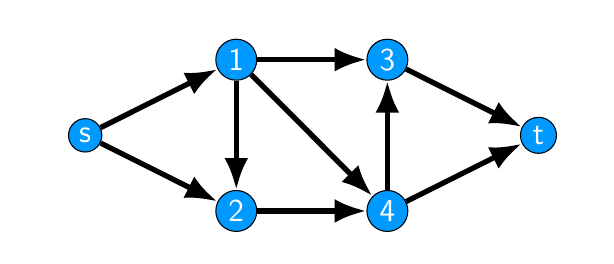
1. **Introduction**

In this essay, we’ll solve 3 basic problems.

* Maximum Network Flow:

In [optimization theory](https://en.wikipedia.org/wiki/Optimization_(mathematics)), maximum flow problems involve finding a feasible flow through a single-source, single-sink [flow network](https://en.wikipedia.org/wiki/Flow_network) that is maximum.

The maximum flow problem can be seen as a special case of more complex network flow problems, such as the [circulation problem](https://en.wikipedia.org/wiki/Circulation_problem). The maximum value of an s-t flow (i.e., flow from [source](https://en.wikipedia.org/wiki/Glossary_of_graph_theory#Direction) s to [sink](https://en.wikipedia.org/wiki/Glossary_of_graph_theory#Direction) t) is equal to the minimum capacity of an [s-t cut](https://en.wikipedia.org/wiki/Cut_(graph_theory))(i.e., cut severing s from t) in the network, as stated in the [max-flow min-cut theorem](https://en.wikipedia.org/wiki/Max-flow_min-cut_theorem).

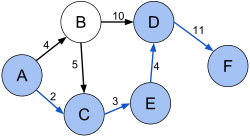


Solving this problem, we will find the longest way from s to t base on our algorithm.

* Shortest Path:

In [graph theory](https://en.wikipedia.org/wiki/Graph_theory), the shortest path problem is the problem of finding a [path](https://en.wikipedia.org/wiki/Path_(graph_theory)) between two [vertices](https://en.wikipedia.org/wiki/Vertex_(graph_theory)) (or nodes) in a [graph](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)) such that the sum of the [weights](https://en.wikipedia.org/wiki/Glossary_of_graph_theory_terms#weighted_graph) of its constituent edges is minimized.

The problem of finding the shortest path between two intersections on a road map may be modeled as a special case of the shortest path problem in graphs, where the vertices correspond to intersections and the edges correspond to road segments, each weighted by the length of the segment.

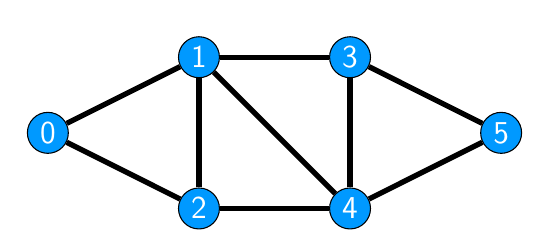


Solving this problem, we will find the shortest way from s to t base on our algorithm. Example in this picture, we’ll find a shortest way from A to F.

* Minimum Spanning Tree:

A minimum spanning tree (MST) or minimum weight spanning tree is a subset of the edges of a [connected](https://en.wikipedia.org/wiki/Connected_graph), edge-weighted undirected graph that connects all the [vertices](https://en.wikipedia.org/wiki/Vertex_(graph_theory)) together, without any cycles and with the minimum possible total edge weight. That is, it is a [spanning tree](https://en.wikipedia.org/wiki/Spanning_tree) whose sum of edge weights is as small as possible. More generally, any edge-weighted undirected graph (not necessarily connected) has a minimum spanning forest, which is a union of the minimum spanning trees for its [connected components](https://en.wikipedia.org/wiki/Connected_component_(graph_theory)).

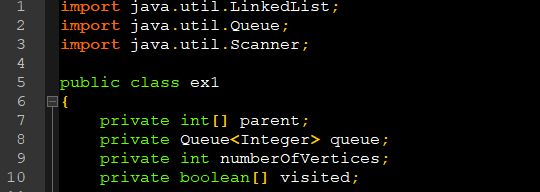
There are quite a few use cases for minimum spanning trees. One example would be a telecommunications company trying to lay cable in a new neighborhood. If it is constrained to bury the cable only along certain paths (e.g. roads), then there would be a graph containing the points (e.g. houses) connected by those paths. Some of the paths might be more expensive, because they are longer, or require the cable to be buried deeper; these paths would be represented by edges with larger weights. Currency is an acceptable unit for edge weight – there is no requirement for edge lengths to obey normal rules of geometry such as the [triangle inequality](https://en.wikipedia.org/wiki/Triangle_inequality). A spanning tree for that graph would be a subset of those paths that has no cycles but still connects every house; there might be several spanning trees possible. A minimum spanning tree would be one with the lowest total cost, representing the least expensive path for laying the cable.

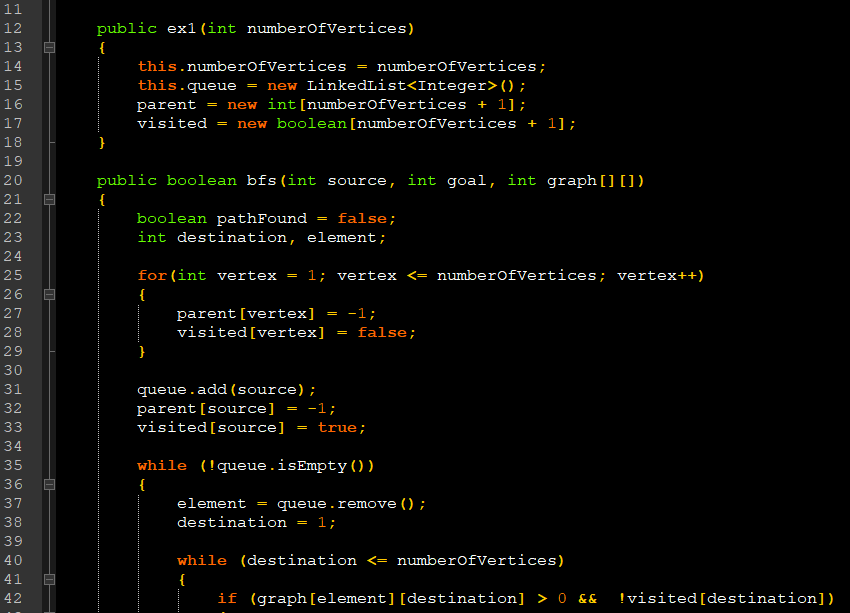


In this algorithm, the graph will not connection. We can move from 0 to 1 and move back from 1 to 0.

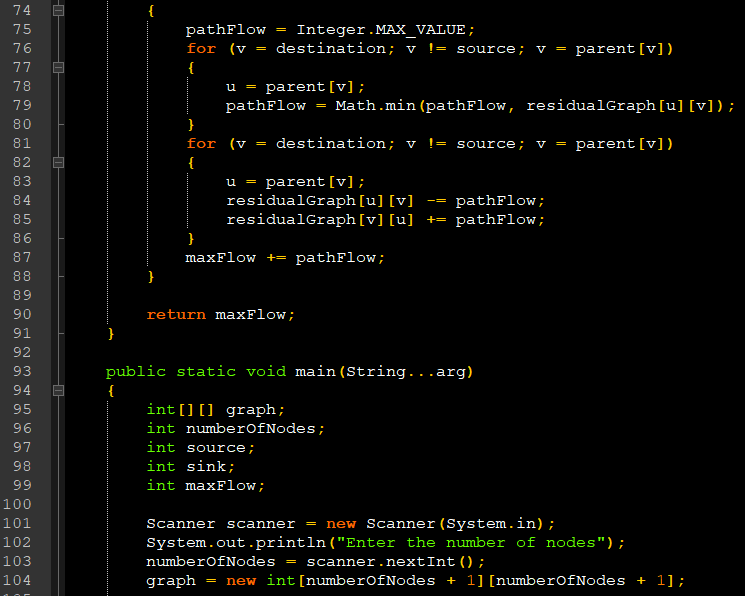
1. Related works

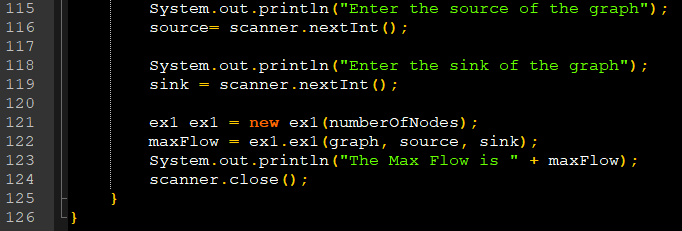
Problem 1: Maximum network flow











1. Approaches
2. Empanelment and results
3. Conclusion